IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 6 December 2022, accepted 22 December 2022, date of publication 2 January 2023, date of current version 5 January 2023.

Digital Object Identifier 10.1109/ACCESS.2022.3233632

RESEARCH ARTICLE

An Efficient Authenticated Elliptic Curve Cryptography Scheme for Multicore Wireless Sensor Networks

ESAU TAIWO OLADIPUPO[1], **OLUWAKEMI CHRISTIANA ABIKOYE**[2], **AGBOTINAME LUCKY IMOIZE**[3,4], **(Senior Member, IEEE)**, **JOSEPH BAMIDELE AWOTUNDE**[2], **TING-YI CHANG**[5], **CHENG-CHI LEE**[6,7], **AND DINH-THUAN DO**[8], **(Senior Member, IEEE)**

[1]Department of Computer Science, The Federal Polytechnic Bida, Bida 912211, Nigeria
[2]Department of Computer Science, Faculty of Information and Communication Sciences, University of Ilorin, Ilorin 240003, Nigeria
[3]Department of Electrical and Electronics Engineering, Faculty of Engineering, University of Lagos, Akoka, Lagos 100213, Nigeria
[4]Department of Electrical Engineering and Information Technology, Institute of Digital Communication, Ruhr University, 44801 Bochum, Germany
[5]Department of Industrial Education and Technology, National Changhua University of Education, Changhua City 500, Taiwan
[6]Research and Development Center for Physical Education, Health, and Information Technology, Department of Library and Information Science, Fu Jen Catholic University, New Taipei City 24205, Taiwan
[7]Department of Computer Science and Information Engineering, Asia University, Taichung City 41354, Taiwan
[8]School of Engineering, University of Mount Union, Alliance, OH 44601 USA

Corresponding authors: Cheng-Chi Lee (cclee@mail.fju.edu.tw), Ting-Yi Chang (tychang@cc.ncue.edu.tw), and Agbotiname Lucky Imoize (aimoize@unilag.edu.ng)

This work was supported in part by the Nigerian Petroleum Technology Development Fund (PTDF) and in part by the German Academic Exchange Service (DAAD) through the Nigerian-German Postgraduate Program under Grant 57473408, and in by the National Science Council, Taiwan, under Grant MOST110-2221-E-018-014 and Grant NSTC111-2622-E-018-004.

ABSTRACT The need to ensure the longevity of Wireless Sensor Networks (WSNs) and secure their communication has spurred various researchers to come up with various WSN models. Prime among the methods for extending the life span of WSNs is the clustering of Wireless Sensors (WS), which reduces the workload of WS and thereby reduces its power consumption. However, a drastic reduction in the power consumption of the sensors when multicore sensors are used in combination with sensors clustering has not been well explored. Therefore, this work proposes a WSN model that employs clustering of multicore WS. The existing Elliptic Curve Cryptographic (ECC) algorithm is optimized for parallel execution of the encryption/decryption processes and security against primitive attacks. The Elliptic Curve Diffie-Helman (ECDH) was used for the key exchange algorithm, and the Elliptic Curve Digital Signature Algorithm (ECDSA) was used to authenticate the communicating nodes. Security analysis of the model and comparative performance analysis with the existing ones were demonstrated. The security analysis results reveal that the proposed model meets the security requirements and resists various security attacks. Additionally, the projected model is scalable, energy-conservative, and supports data freshness. The results of comparative performance analysis show that the proposed WSN model can efficiently leverage multiprocessors and/or many cores for quicker execution and conserves power usage.

INDEX TERMS Multiprocessor, multicore, wireless sensor, encryption, chosen plaintext attack, chosen ciphertext attack, IND-CPA, IND-CCA.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are self-configured and infrastructure-free wireless networks that track environmental or physical conditions [1]. Examples of such physical and environmental conditions include temperature, sound, vibration, pressure, motion, or pollutants. WSNs facilitate the sending of relevant information via the network to a centralized location or sink where the data can be observed and analyzed. WSNs are being used in the military for monitoring

The associate editor coordinating the review of this manuscript and approving it for publication was Arun Prakash.

purposes [2] as well as in other works of life (medicine, agriculture, sport, etc.) [3], [4], [5]. WSNs are composed of several wireless sensors (WS) that are capable of communicating with other WS using radio signals. WSN is one of the most intriguing aspects of the wireless network research field [6]. A typical WSN comprises a randomly distributed WS (node) in a particular area [7], [8]. A WSN node has four major components: sensing, processing, transceiver, and power unit.

Because WSNs are battery-powered, they eventually die after a certain amount of time [9]. The basic advantages of networking include resources (e.g., memory and processor) and information sharing, supply of neighborhood amenities while maintaining centralized management, even distribution of work, and improved communication facilities [10]. WSN, as a type of network, is therefore capable of benefitting from all these advantages. The major operations among the nodes in WSN include communication between nodes and Base Station (BS), and data processing by each node. If the lifetime of WSN is to be enhanced, there should be optimization to these WSN operations [11], [12]. Existing techniques of extending network lifetime in literature incorporate a number of important limitations and routing techniques for increasing energy efficiency [13], [14], [15] and clustering [16], [17], [18], [19] for ensuring proper network operation. None of these methods talks about improving the processing power of the sensor to achieve a reduction in power consumption during these operations. Because of advancements in embedded systems, application software must view the general-purpose processor element using a multiprocessing paradigm for this processor to share in the benefits of higher performance and low power [20]. A multicore processor, according to [21], is an integrated circuit (IC) to which two or more CPUs have been connected to improve performance, lower power usage, and more effectively handle numerous tasks at once. Energy consumption which is an integral of the power consumption over the time needed to operate, can be calculated as the multiplication of power consumption and time. This energy consumption determines how much energy is drained from the battery to complete a specific task [22]. Since multicore systems are capable of increasing the throughput [23], the time for completing the task is reduced which means the energy consumption is reduced. Hence, the processing unit of WS in WSN can be replaced with multicore processors for higher performance and low power consumption.

The literature is replete with parallelized schemes for Elliptic Curve Point Multiplier (ECPM) [24], [25], [26]. This is probably because researchers believe that the complexity of the Elliptic Curve Cryptosystem is mainly dependent on Elliptic Curve Scalar Multiplication (ECSM) [27]. Little or no attention has been given to how the ECC algorithm's entire operations involved in the encryption/decryption processes can be parallelized. With the proliferation of multiprocessor and multicore devices aimed at increasing the processing speed, software design considerations should be able to include techniques that are capable of efficiently distributing the software functionality across these computing resources [28]. Therefore, this work proposes a new WSN model that employs multicore processors and an optimized ECC algorithm for the parallelization of encryption/decryption processes among the processor cores.

A. MOTIVATION

The major challenges facing the implementation of the encryption/decryption process are the issue of how the conversion of plaintext data on the elliptic curve to point (mapping) and the conversion of a point on the elliptic curve to plaintext (reverse mapping) can be securely done [29], [30]. Several attempts, such as [31] and [32] are both vulnerable to cryptanalysis attacks and are not able to distribute encryption/decryption functionality across processors and/or cores in multiprocessor or multicore systems. For example, the application of Cipher Block Chaining (CBC) mode by [29], [33] makes the systems not able to distribute encryption/decryption functionality across processors or cores in multiprocessor or multicore systems. In addition, these systems are vulnerable to Man-in-the-Middle-Attack (MIMA), Chosen Plaintext Attacks (CPA), and Chosen Ciphertext Attacks (CCA) [34].

To the best of our knowledge, there has not been a single research work that reports an ECC-based encryption/decryption scheme that adopts a domain decomposition programming model to achieve parallelization.

B. CONTRIBUTION

This paper proposes a model for WSNs where each node is built with multicore processors. An ECC-based encryption/decryption scheme for parallelizing encryption/decryption operations within a node is proposed. The ECC-based encryption/decryption scheme adopts a domain decomposition programming model where data to be encrypted/decrypted are broken into smaller components that can be independently encrypted/decrypted. This approach removes data dependency in an attempt to ensure the security of data during the iterative procedure of encryption/decryption processes. This data dependency removal makes the scheme suitable in any multiprocessing system as the iterative process can be shared among the processors in multicore or multiprocessor systems. Elliptic Curve Diffie-Helman (ECDH) is used for key exchange while Elliptic Curve Digital Signature Algorithm (ECDSA) is applied to achieve digital signing and signature verification in the scheme in order to ensure confidentiality, integrity, and availability of the data in the proposed scheme.

The contribution of this paper includes:

i. Introduction of multicore wireless sensors into a hierarchical clustered wireless sensor network for the speed of execution in WS and a further reduction in energy consumption of WSN.
ii. Modification of the existing ECC-based algorithm to suit parallel encryption/decryption processes to efficiently maximize multicore processors in the WSN.

iii. Introduction of mutual authentication between any two nodes to avoid communication among legitimate and stranger nodes.

iv. Introduction of forward and backward secrecy into WSN such that a new node cannot decrypt messages before joining the network and an old member of the network that left the network should not be able to decrypt the messages communicated after leaving the network. Messages sent to a particular node through multi-hop cannot be decrypted by intermediate nodes.

The remaining sections of this paper are structured as follows. Section II presents some related work on multithreading, multiprocessing, and multicore. The materials and methods used for the study are covered in Section III. Section IV outlines the security analysis of the proposed scheme. Finally, Section V concludes the paper and provides future perspectives.

II. LITERATURE REVIEW

A review of the concept of multiprocessing/multicore and Elliptic Curve Cryptography as well as related works, are handled in this section.

A. MULTIPROCESSING, MULTITHREADING, AND MULTICORE

The concept of multithreading, multiprocessing, and multicore was introduced into processor design to enhance the processor's overall functionality such that the processing time and power consumption of any application software that exhibits concurrency execution will be reduced [20]. In multithreading technology, the processor performance is increased by utilizing the period of dormancy in a uniprocessor design which normally happens as a result of pairing a fast processor with slower memory. The goal is to achieve the highest performance possible from the execution of multitasking software without adding any of the software complexity costs associated with multithreading. Multiprocessing is a method that uses the notion of duplicating a processor's design. Using multicore CPUs is one way to boost processor performance while remaining within the realistic constraints of semiconductor design and production [39].

A widely accepted classification of multiprocessor is that of Michael Flynn who based his classification on two orthogonal parameters: the instruction stream and the data stream. In Flynn's multiprocessor naming scheme [40], to categorize a machine, it was determined whether it had a single or numerous data streams and instruction streams. Flynn's taxonomy scheme leads to four possible combinations which include: SISD: single instruction, single data; i.e., a conventional uniprocessor. SIMD: single instruction, multiple data MISD: multiple instructions, single data; MIMD: multiple instructions, multiple data; SMPs, clusters. This is the most common multiprocessor. It has been argued that Flynn's taxonomy has some fundamental problems. One of the problems has to do with the MIMD category in which the processor architecture with many processors is grouped as MIMD without considering how memory is connected to the processors. Attempts have been made to categorize MIMD into two major parallel architectural paradigms: SMP (symmetric multiprocessors) and MPP (massively parallel processors). While SMP machines share a memory, MPP machines do not. SMP can further be divided into two: uniform memory access (UMA) and non-uniform memory access (NUMA). All the processors in UMA have equal access to physical memory while memory access in NUMA machines is both non-uniform and has physically separate memories, attached locally to each processor.

B. CONCEPT OF ELLIPTIC CURVE CRYPTOGRAPHY

A brief introduction to elliptic curve cryptography is given in this section. The key benefit of elliptic curve cryptography is that it offers improved security with reduced key size. For instance, a 160-bit elliptic curve offers the same security as that of a 1024-bit RSA. On the other hand, the use of symmetric cryptography algorithms such as Advanced Encryption Standard requires that a secure means of exchanging the key between communicators is established. This may expose the system to a MIMA attack. With ECC, ECDH algorithms are handy for key exchange without being susceptible to MIMA attacks. In mathematics, elliptic curves are described by (1), which is normally described as the Weierstrass equation.

$$Ax^3 + Bx^2y + Cxy^2 + Dy^3 + Ex^2 + Fxy + Gy^2 + Hx + Iy + J = 0, \tag{1}$$

In cryptography, an elliptic curve E over a finite field GF(Fp) is the simplified form of the Weierstrass equations which is defined by (2) [35].

$$y^2 = (x^3 + ax + b) \ mod \ p \tag{2}$$

where a, b, and p satisfy the given description in Table 1. The set of points on these simplified elliptic curves and a special point ∞ (infinity) form the Abelian group under addition. The identity element of the group is ∞. The operations on elliptical curves in cryptography are point addition and point doubling [36], [37], [38]. Point doubling forms the building blocks for point scalar multiplication.

1) POINT ADDITION

Let $P(x_1, y_1)$ *and* $Q(x_2, y_2)$ be two points on E. Addition of P and Q given by equation (3) will result in a point R which is also a point on E or a special point ∞.

$$R = P + Q = \begin{cases} \infty, & if\ x_1 = x_2\ and\ y_1 = -y_2 \\ & OR\ x_1 = x_2\ and\ y_1 = y_2 = 0 \\ P, & if\ Q = \infty \\ Q, & if\ P = \infty \\ (x_3, y_3), & otherwise \end{cases} \tag{3}$$

where

$$x_3 = \begin{cases} \lambda^2 - x_1 - x_2, & if\ P \neq \pm Q \\ \lambda^2 - 2x_1, & if\ P = Q \end{cases}$$

$$y_3 = \lambda (x_1 - x_3) - y_1$$

and

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, & \mathit{if}\ P \neq \pm Q \\ \dfrac{3x_1^2 + a}{2y_1}, & \mathit{if}\ P = Q \end{cases}$$

2) POINT SCALAR MULTIPLICATION

Let M be a random point on the elliptic curve. The operation of multiplying M by a constant scalar k is done by repetitive addition. To achieve encryption and decryption using ECC, k|M| plays a vital role in the exponentiation operation.

k[M]= M + M + M +···+M, k times i.e, 5M = M+M+M+ M+M (additions). Point doubling is very useful when it comes to a reduction in the number of additions to be carried out.

M+M = 2M(Doubling)
2M+2M = 4M(Doubling)
4M+M = 5M(addition).

C. RELATED WORKS

For wireless multimedia sensor networks (WMSN), an effective system for authentication schemes and identification was created by authors in [38]. The model's technique uses a combination of modified ECC and digital hashing to give authentication. A node that wants to transfer data to another node creates a shared key utilizing both the private key and the public key of that node. The shared key is used to encrypt the message using the sender's public key and its private key, the receiving node likewise creates the shared key. The system proved to be impenetrable to all recognized attackers. However, the costly assumption of this scheme is based on the fact that the sending node is a legitimate member of the network. The receiving node has no formal way of authenticating the sender before it establishes communication with the sending node. Hence, receiving node may be communicating with a privileged stranger who has information about the elliptic curve used in the communication. Likewise, the same costly assumption was made by [6] in the establishment of communication between two nodes.

The author in [41] suggested the encryption algorithm Elliptic-Curves-Diffie Hellman-Rivest-Shamir-Adleman (ECDH-RSA). The suggested encryption is used under the chessboard clustering routing method (CCRM). The nodes' energy use was regulated by the CCRM. By combining a small number of highly effective max-end sensors in addition to many highly effective min-end sensors, to accomplish efficiency and scalability goals. Because it can offer high protection with a small key size, and the exchanging of public and private keys is done using ECDH. The findings of the proposed model's computation demonstrated that the method can enhance WSN life by (47%, and 35.7%) when compared with an adaptive clustering superstructure and safe low-energy (Sec-LEACH and SL-LEACH) approaches.

The work in [42] presented key management schemes in hierarchical WSNs. The following three types of keys are supported by this scheme for creation and updating: network key, group key, and pairwise key. A network key needs to be first produced to encrypt messages and authenticate additional nodes, and possessed by each node. All nodes in the same cluster head (CH) share the same group key, and a specific pair of network nodes share a pairwise key. The network's nodes can be added and revoked using this approach. The approach relies on a hierarchical structure that gives the network flexibility and scalability.

In sensor networks with little computational and communicational load, [43] proposed an authentication strategy to fend against a variety of weak attacks. Registration, node validation by cluster leaders, mutual authentication, and distribution of the generated secret keys are the four processes used to address the key management and access control challenges. A new key is generated and distributed for every session to thwart replay attempts. Engineered cementitious composites (ECC), which are resistant to different attacks, are the foundation upon which the proposed authentication system is built. Their experimental findings indicated that less energy is needed for effective authentication. With increased security, it also became better with delays and traffic congestion.

The ECC-based scheme proposed for edge computing and the internet of things by [33] is vulnerable to MIMA, CPA, and CCA attacks [34]. In addition, the application of CBC makes it difficult for the scheme to apply a domain decomposition programming model which can efficiently maximize the resources when used for parallelization in multiprocessor/multicore systems.

In the work of [44] and [6], the Elliptic Curve Diffie-Helman technique for key exchange was adopted. This means that communication can be established between any two nodes once they can establish a connection. This approach gives room for a privileged stranger node who has the idea of the id of a node and the elliptic curve used for secure communication. Such a privileged stranger node can send its public key to a legitimate node in WSNs and since the legitimate node does not have a way of verifying the source, the connection is established, and then the secret may be leaked by this means. The following illustrates the scenario:

Let A and B be legitimate nodes in WSNs and let S be a stranger node that is not part of the network but has the privilege of information about the elliptic curve and address of A or B or both. Let $pr_{Na}, pr_{Nb}, pr_{Ns}$ be private keys of A, B, and S and $pub_{Na}, pub_{Nb}, pub_{Ns}$ be public keys of A, B, and S, respectively. For A and B to communicate: A computes the share secrete key $pr_{Na} \times pub_{Nb}$ and B also does the same to get $pr_{Nb} \times pub_{Na}$. For S and A to communicate: S computes $pr_{Ns} \times pub_{Na}$. A, however, does not have a way to verify whether S is a member of the network or not. So, A also computes $pr_{Na} \times pub_{Ns}$. Under this circumstance, A is talking to a stranger S, without knowing.

From the review works, the following problems are identified:

i. A node in the existing WSN models is using uniprocessors. Hence, parallelization of iterative operations among multiprocessors/multicores is not possible
ii. Existing ECC applies Cipher Block Chaining (CBC) mode to ensure confidentiality. However, this approach prevents the use of a domain decomposition programming model which can efficiently maximize the resources when used for parallelization in multiprocessor/multicore systems
iii. authentication among the communicating parties cannot differentiate between legitimate members and strangers.
iv. Existing ECC encryption/decryption algorithms are still susceptible to CPA, CCA, and MIMA.

III. MATERIALS AND METHODS

A. THE PROPOSED WSN MODEL

The review works established the fact that researchers have done a lot to provide WSN models with different cryptographic algorithms to ensure secure and authenticated communications among the nodes and integrated routing methods to cut down on energy usage and therefore increase the lifespan of WSNs. However, the application of multicore wireless sensors with hierarchical clustering protocol of WSNs' power usage and longevity have not been explored in any depth. This gap and others as identified in the analysis of the latter part of the previous section have to be breached. In this paper, the identified gaps are addressed.

The suggested WSNs model is predicated on the following premises:

- The network has three different types of nodes: Base Station (BS), Cluster Head (CH), and ordinary node (N).
- All the nodes have multicore/multiprocessing units
- BS is capable of connecting directly to every network connection
- When a node enters or departs the network, BS updates a database that contains records of all the nodes' IDs.
- The network's sensor nodes are all stationary.
- In the cluster, the CHs are responsible for data transmission, management, and node administration, and they are only one hop away from the BS.
- The sensor node N communicates location-specific data to the CHs and can be accessed from the CH with a single hop or several hops.
- Each CH has a Global Positioning System (GPS).
- An attacker needs a time T to compromise any node.
- A timestamp named "T" that is included in every message sent and received ensures the accuracy of the data.
- The network has a hierarchical structure as depicted in Figure 1.

The proposed WSN scheme presented in this section is divided into seven subdivisions: the generation of initialization parameters, node distribution, cluster forming, creating a shared key, Node removal, Updating the shared key, and Secure data transmission.

Generation of Initialization Parameters: This stage assists in producing the shared secret key (ssk). This ssk is used by any of the nodes for secure message transmission. Table 1 provides a list of the parameters used in the proposed approach and a description of the notations used in this work.

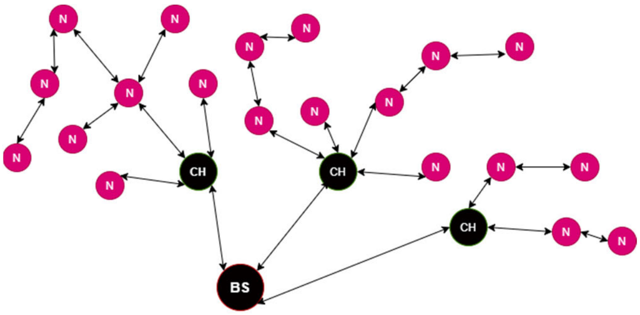

FIGURE 1. The network structure of the proposed WSN model.

Algorithm 1 describes the operations that take place during the initialization procedure. The EC encryption and decryption algorithms have different EC curves that can be used during communication between any nodes or between any node and BS. This is decided when the transmission is to take place in the network.

Algorithm 1 Initialization Procedure

```
START
BS assigns $id_{BS}$ to itself
FOR i = 1 to k number of CHs
    BS assigns unique $id_{CHi}$ to CH[i]
ENDFOR
FOR i = 1 to n number of node Ni
    BS assigns unique $id_{Ni}$ to Ni[i]
ENDFOR
LOAD EC encryption and decryption algorithms into
BS, CHi[1..k] and Ni[1..n]
STOP
```

B. NODE DISTRIBUTION

BS creates a variety of IDs for sensor nodes (id_{BS}, id_{CHi} and id_{Ni}) in the network. Several nodes, say k wireless sensors, are uniformly dispersed randomly within the area say (k × k) m^2 [42], [45]. Each of the sensor nodes (BS, CHi, and Ni) can confidentially communicate with and mutually authenticate other nodes.

C. NODES CLUSTERING

The method of clustering involves dividing a sensor network's sensing field into several clusters. A leader known as the cluster head is chosen by each cluster [46], [47]. Clustering techniques are introduced into WSNs to ensure the longevity of their lifetime by reducing their power consumption [6] and increasing their scalability [48]. Cluster partition for homogenous WSNs which has been comprehensively researched [49], [50], [51] is employed in the proposed WSNs model. Node clustering begins after the deployment of the sensors to the area. The CHs are chosen both randomly [52]

TABLE 1. Description of the notations used in the proposed scheme.

Notation	Description
id_{BS}	Base station identification number
id_{Ni}	Node identification number
id_{CH}	Cluster Head identification number
$y^2 = x^3 + ax + b \bmod p$	EC equation
p	The large prime number obtained from chosen EC
a,b	Constants of EC such that $4a^2 + 27b^2 \ mod \ p \neq 0$
G	the base point which produces the subgroup of the elliptic curve that has a large prime as its order.
n	Order (number of the point) of the subgroup. nG = 0, prime n is large
h	the cofactor of the subgroup which is the ratio $\frac{\lvert E\rvert}{\lvert E_p\rvert} = \frac{order\ of\ elliptic\ curve\ E}{order\ of EC \text{ over a prime field } E_p}$. h should be small $h \leq 4$, preferably, h =1
pr_{BS}	Base station private key
pr_{Ni}	Node private key
pr_{CHi}	Cluster head private key
pub_{BS}	Base station public key $(pr_{BS} \times G)$
pub_{Ni}	Node public key $(pr_{Ni} \times G)$
pub_{CHi}	Cluster head public key $(pr_{CHi} \times G)$
H_s	The hash function used by the base station
nList	Node list containing reference id's $(id_{BS}, id_{Ni}, id_{CH})$
HASH	Signing cipher message Cm hash function
ssk	Shared secret key
SNP	Share node point = (ssk x G)
PRK	Private random key (bit size is dependent on the type of EC)
K	A random integer is chosen from (1, p-1)
CM	The ciphertext (all encrypted points)
$\oplus$	Exclusive OR operation
+	Addition operation used for ECC encryption process
T	Time stamp

and based on the use of some criteria such as the CH signal that was the greatest [53], [54], distance, and residual energy of the CH [55]. Energy-aware fuzzy clustering algorithm (EAFCA) proposed by [56] is employed in the formation of the clusters and the choice of CHs. The mechanism used by EAFCA involves sending a beacon signal from the BS to the other sensor nodes. A sensor node can determine its distance from the BS using the received signal strength. The sensor network is then used to elect a group of tentative cluster heads (TCHs) for a particular portion of the entire network. To all of the sensor nodes, a threshold "t" is calculated and sent. Each sensor node produces a random number, which it then compares to the received threshold value. The sensor node identifies itself as the cluster head if the generated value exceeds the threshold. Otherwise, it turns into a regular sensor node. In order to select CHs from TCHs, fuzzy logic is used. The fuzzy logic is based on the following three parameters:

Remaining residual energy (RRE): Residual energy is energy left in a node after performing its processing and data-transferring functions [57]. RRE is expected to be higher for eligible CH because it is heavily engaged in intra-cluster and inter-cluster data traffic.

Node degree at its 2-hop coverage: This parameter represents the total number of neighbors that are located in the 2-hop distance from the tentative CH and is calculated using equation (4). A tentative cluster head should have a higher value for this parameter to become a permanent cluster head.

$$2 - hop\ node\ degree = \frac{\left|S_{2-hop-nbr(i)}\right|}{number\ of\ nodes} \tag{4}$$

where $S_{2-hop-nbr[i]}$ gives the total number of neighbors for the tentative cluster head in its 2- hop coverage.

The centrality of CH: Node centrality is calculated using equation (5). A CH should have low node centrality to reduce energy consumption during the data aggregation and flooding processes.

$$Node\ centrality = \frac{\sqrt{\frac{\left(\sum_{S_{2-hop-nbr(i)}} d^2(i,j)\right)}{\left|S_{2-hop-nbr(i)}\right|}}}{A} \tag{5}$$

where d(i, j) represents the distance between nodes "i" and "j" in which node j is a member of the set 2-hop-nbr and A represents the area of the network.

Communication between CH and BS is achieved through a single hop while communication between CH and a member node can be either single or multi-hop. As a measure to reduce the energy consumption of a CH, new nodes are selected as CH after a certain interval of time [58]. By routing CHs, energy consumption among sensors can be averaged [9], [59].

D. CREATION OF SHARED KEYS

The shared secret key is generated and kept by BS. This makes it possible for WSN nodes to successfully decrypt the cipher text. Therefore, utilizing its H_s, id_{BS}, and PRK, BS generates the initial ssk. Algorithm 2 illustrates the key generating procedure.

Algorithm 2 Procedure for Creating an Initial Shared Secret Key

```
Input: $H_s$, $id_{BS}$, PRK
Output: ssk
1.  START
2.  ssk = $H_s\ (id_{BS}) \oplus PRK$
3.  $initial\ shared\ secrete\ key \leftarrow ssk$
4.  STOP
```

E. UPDATING OF THE SHARED SECRET KEY

Updating of shared secrete key is done by all the nodes (BS, CHi, and Ni), this is done to ensure both forward and backward secrecy. Each time a node joins/leaves the network, the shared secret key has to change. To ensure that no new node can decrypt ciphertext sent before it joined the network, ssk is altered before the new node is allowed to join. Likewise, when a node leaves the WSN, ssk is altered so that any node that leaves the network will not be able to decrypt the ciphertext that is sent after departure using ssk. The update of ssk by an existing node in WSN is carried out using equation (6)

$$\text{ssk} = H_s\left(id_{Ni}\right) \oplus ssk \tag{6}$$

If a node Ni joins the network, the BS notifies all active nodes of the newly joined node's hashed ID $H_s\ (id_{Ni})$. Each node then changes ssk using Equation (6) while simultaneously updating its node list nList. The same equation is also used by the BS to update its ssk. After the update of the ssk and nList by BS and all the existing nodes, BS sends ssk as well as a list of all the hashed IDs for network nodes (the new node's hashed ID is the only exception, albeit). Algorithm 3 outlines the steps necessary for a new node Ni to join the WSN.

Algorithm 3 Procedure for Updating ssk, snp, and nList When a Node Joins WSN

```
Input: $id_{Ni}$, $H_s$, ssk
Output: Updated ssk, snp and nList
Start
New node requests join ($id_{Ni}$);
BS generates $pr_{BS}$ and $pub_{BS}$
BS broadcast $pub_{BS}$
Current nodes generate $pr_{Ni}$ and $pub_{Ni}$
Current nodes broadcast $pub_{Ni}$
BS computes SNP = snp x $pr_{BS}$ x $pub_{Ni}$ //for each node Ni
BS broadcast joinednode = $H_s\ (id_{Ni})$ + snp //using the corresponding snp of each node Ni
Current nodes compute snp = snp x $pr_{Ni}$ x $pub_{BS}$
Current nodes compute joinednode = joinednode - snp
BS and current nodes update key ssk = $H_s\ (id_{Ni}) \oplus$ ssk
BS and current nodes update shared point snp = ssk $\times$G;
BS sends ssk and nList to new node;
BS and current nodes update their nList
Stop
```

A node that is exiting the network should indicate so through its CH to the BS. However, there may occur some exceptional cases where a node exits the network without notifying the BS. The CH is saddled with the responsibility of checking the active nodes by sending dummy messages and receiving a response. Any node that fails to respond to this message is considered inactive or had left the network. So the CH reports such a node as the node that is not on the network. Every current node on the network receives the id_{Ni} of the node departing the network from BS through their CH. Each node updates its ssk by utilizing equation (3). The necessary hashed IDs are simultaneously deleted from the network list of the presently linked nodes. The BS additionally updates ssk using equation (3) and removes the related hashed ID from its node list. Algorithm 4 provides instructions for carrying out a node's departure operation.

Algorithm 4 Procedure for Updating ssk, snp, and nList When a Node Leaves WSN

```
Input: $H_s$, $id_{Ni}$, ssk
Output: Updated ssk, snp and nList
Start
A node sends leave ($id_{Ni}$)
BS generates $pr_{BS}$ and $pub_{BS}$
BS broadcast $pub_{BS}$
Current nodes generate $pr_{Ni}$ and $pub_{Ni}$
Current nodes broadcast $pub_{Ni}$
BS computes snp = snp x $pr_{BS}$ x $pub_{Ni}$ //for each node Ni
BS broadcast leave node = $H_s\ (id_{Ni})$ + snp //using the corresponding snp of each node Ni
Current nodes compute snp = snp x $pr_{Ni}$ x $pub_{BS}$
Current nodes compute leavenode = leavenode - snp
BS and current nodes update key ssk = $H_s\ (id_{Ni}) \oplus$ ssk
BS and current nodes update shared point snp = ssk $\times$G;
BS and current nodes update their nList
Stop
```

F. SECURE DATA TRANSMISSION

Maintaining security requirements – confidentiality, integrity, authentication, authorization, availability, and non-repudiation - during data transmission is of paramount importance for reliable and dependable communication [60], [61], [62], [63]. To ensure the confidentiality of communication, the message is encrypted by the sending node before transferring it to the receiving node. This stops unauthorized parties from accessing the data being sent [57], [58]. To ensure the integrity of the encrypted message being sent, the sending node appends the signature before sending the message to the receiving node. The receiving node authenticates the integrity of the received message before going further to decrypt the received message. Any third party who wants access to the transmitted information must have access to the private key of the recipient and snp otherwise authorization to access the information is not given. The receiving node can confidently identify the source of the received message and the source cannot deny it through the use of the signature and verification of the signature. Hence non-repudiation characteristics are also achieved. The hierarchical model of data transmission has two unique aspects. Member nodes first deliver data directly or through a multi-hop mechanism to their CH. CHs transmit data, in a single hop, to the BS. The above security characteristics are enforced in secure data transmission using four processes: encryption of the message, signing of the encrypted message, verification of the encrypted message, and decryption of the verified encrypted message.

1) ENCRYPTION OF THE MESSAGE

The initial task is to determine how many cores are present in the sensor where the encryption/decryption operation will take place. This programmatic determination of the number of cores in the sensor will be used for decision-making on how data is to be shared among the cores during the execution of the iterative procedure. The choice of EC in the Elliptic Curve Cryptosystem (ECC) determines the key length and the size of the block that can be handled by the ECC. The data to be encrypted is divided into blocks after the generation of the key from the chosen EC. Each block is converted to binary digits. Algorithm 5 explains how plaintext is broken up into several blocks.

Algorithm 5 Conversion of Plaintext Into a Set of Blocks

```
Input: Plaintext, p
Output: a set of blocks
Start
LET M = number of characters in Plaintext
COMPUTE the size of each block from the chosen elliptic
  curve, $N = FLOOR\left(\frac{p-8}{8}\right)$
COMPUTE the number of blocks in the
  plaintext $numblocks = CEIL\left(\frac{M}{N}\right)$
LET B = [ ]
t = 1
FOR i = 1 TO numblocks
  B[i] = Plaintext[t: t+ N]
  t = t + N
ENDFOR
$set\ of\ blocks \longleftarrow B$
STOP
```

The generated shared secret key(ssk) is multiplied by G to obtain a shared node point (snp). Private key pr_{NiA} and public key pub_{NiA} are generated by the sending node. pr_{NiA} is retained by the sender while the pub_{NiA} is transmitted through the wireless channel. The receiving node B also does the same to generate private and public keys pr_{NiB} and pub_{NiB} sends out pub_{NiB} while keeping pr_{NiB} secret. This principle uses ECDH for key exchange. To avoid communication with alien nodes, node A multiplies the generated snp by $(pr_{NiA} \times pub_{NiB})$. The x coordinate of the result is then used to generate an array of secure random numbers whose size is the number of blocks of the data to be encrypted. The data blocks and the array of random numbers are then grouped into several processors. For example, if a WSN node has 4 cores P1, P2, P3, and P4 and the number of blocks is n then each core will be assigned n/4 blocks to be encrypted. If n is not divisible by the number of processors, then the last processor will be assigned the remaining blocks for encryption. That is, the distribution of encryption operations to each processor will be as follows:

P1 will be assigned blocks 1…n/4,
P2 will be assigned blocks n/4 + 1…n/2,
P3 will be assigned blocks n/2 + 1…3n/4, and
P4 will be assigned blocks 3n/4+1…n.

The secure random numbers generated from the x coordinate of snp will also be shared among the cores in the same proportions. After the encryption process is completed by each core, the result of the encrypted blocks from each core will be combined in the order P1, P2, P3, and P4, respectively. The signing of the encrypted data is then carried out before transmission to the recipient. Figure 2 illustrates how the encryption and signing processes are carried out by a node in WSN. Each core simultaneously reshuffles the binary digit of each block assigned to it to secure it from encryption attacks, maps the block to a point on an elliptic curve, and encrypts the block. Algorithm 6, Algorithm 7, and Algorithm 8, respectively describe how these procedures are carried out.

2) THE SIGNING OF THE ENCRYPTED MESSAGE

The secrecy of the message is accomplished by encrypting the message that will be sent. However, the integrity of the message is not achieved because a third party can modify the encrypted message without the knowledge of both the sending and receiving nodes. Therefore, to be convinced of the integrity of the message, the sending node A appends a signature to the encrypted message using its private key pr_{NiA}, which depends on ECDSA. The description of how the sending node A appends its signature to the encrypted message is given in Algorithm 9.

3) SIGNATURE VERIFICATION OF THE SIGNED AND ENCRYPTED MESSAGE

The receiving node B does not just start the decryption of any received message. It first verifies the authenticity of the received message. To verify that the received message originated from the sending node A, Node B validates the message it has received using pub_{NiA} of node A. Algorithm 10 describes the procedure taken by receiving node B in verifying the received message $M_{received}$.

4) THE DECRYPTION OF THE VERIFIED RECEIVED MESSAGE $M_{received}$

If $M_{received}$ is verified to have originated from the sending node A. then it is confirmed that the actual message M_{sent} from Node A is the $M_{received}$ by the receiving node B. Node B then extract the C_m component of the received message (since $(M_{sent} = (C_m, t_o,) = M_{received})$. The C_m is then subjected to a decryption algorithm by the receiving node B. The process of decryption also takes the advantage of the fact that the WS has a multicore processor. Hence the decryption process is shared among the cores. This process increases the speed of execution and also reduces the energy consumption of the node. The decryption process as shared by the cores in a WS is depicted in Figure 3.

The decryption process involves the extraction of the encrypted points C_m from the verified message $M_{received}$. To be sure that the node is not alien to the network, ssk is multiplied by G to obtain snp. This snp is multiplied by $(pr_{NiB} \times pub_{NiA})$. The points in C_m are counted and the x coordinate of the result of multiplying snp by

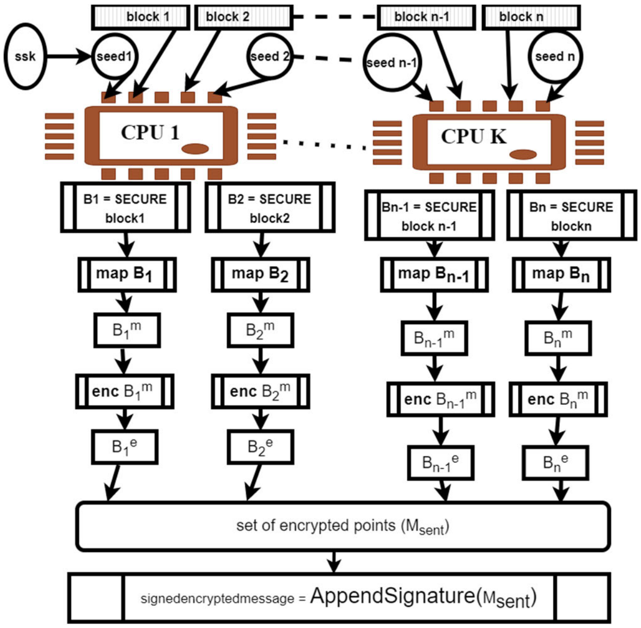

FIGURE 2. Encryption and appending of signature by a node in WSN.

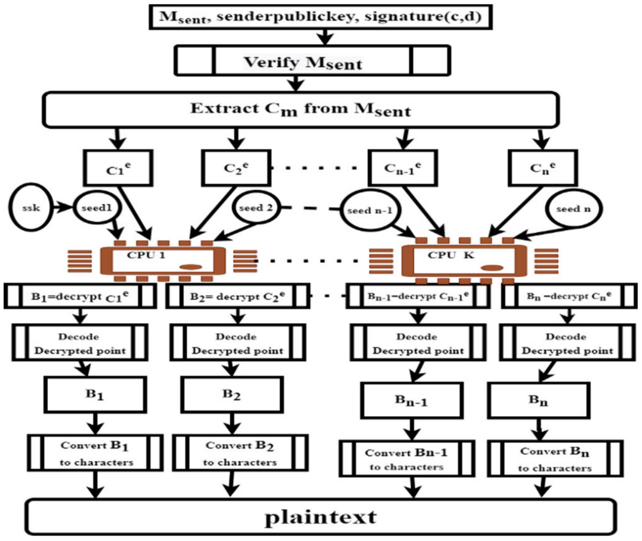

FIGURE 3. Verification and decryption processes by a node in the proposed WSN model.

($pr_{NiB} \times pub_{NiA}$) is used to generate an array of seeds whose size is equal to the number of counted points in C_m. The points and the seeds in the array of seeds are grouped based on the number of cores in the sensor, then the processors can simultaneously decrypt each point in the group of points given to them. Algorithm 11 gives the decryption of how points are decrypted.

After the points are decrypted, it is necessary to decode the decrypted points and convert the decoded points to plaintext. Algorithm 12 and Algorithm 13 outline the procedure for decoding and conversion of the decoded points to plaintext, respectively.

Algorithm 6 Procedure for Protecting Blocks From Encryption Attack

```
Input: a set of blocks (in binary digits), ssk, G
Output: a set of secured blocks (in binary digits)
START
$Ptext \leftarrow set\ of\ blocks$
LET C = LENGTH(Ptext)
COMPUTE $snp = ssk \times G$
COMPUTE $snp = snp \times pr_{NiA} \times pub_{NiB}$
GENERATE C seeds using the x coordinate of snp
R = [ ]
FOR i = 1 to C
  Pblock = ptext[i]
  S = seeds[i]
  Lpt = LENGTH(Pblock)
  LET pattern = GENERATE array of random integers(1..Lpt) using S as a seed
  INITIALIZE message as empty matrix
  LET k = 1
  FOR col = 1 TO Lpt
    LET x = pattern[col]
    LET message [k] = Pblock[x]
    LET k = k + 1
  ENDFOR
  R[i] = message
ENDFOR
$set\ of\ secured\ blocks \leftarrow R$
STOP
```

IV. SECURITY ANALYSIS OF THE PROPOSED SCHEME

Maintenance of the Confidentiality, Integrity, and Authenticity (CIA) of the data being shared are major challenges to the deployment of wireless communication systems [64], [65], [66]. Other issues that are of paramount importance for consideration in the development of the WSNs scheme are data freshness in which WSNs ensured that outdated data are not replayed by malicious nodes in the network. It is also important to ensure that WSNs' routing protocols support scalability which ensures that the performance of the networks is not degraded as the size of the network is increasing [66], [67], [68]. A good WSN model should be resistant to different attacks which normally take the advantage of the vulnerable nature of wireless networks to break its security. Examples of these attacks include eavesdropping, Sybil attack, Man-in-the-Middle attack, Replay attack and denial of service attacks. It is also expected that the cryptography algorithm used in protecting the message being shared among the network members should not be vulnerable to cryptanalysis attacks. Primitive attacks such as Known plaintext Attack (KPA), Chosen Plaintext Attack (CPA), Chosen Ciphertext Attack (CCA), Adaptive Chosen Ciphertext Attack (CCA-2) and Authenticated Adaptive Chosen Ciphertext Attack (CCA-3) should not be able to break the security of the data being protected. The performance analysis of the proposed scheme is evaluated based on various aspects

Algorithm 7 Mapping of Secured Blocks to Points on the Elliptic Curve

```
INPUT: secured blocks
OUTPUT: mapped points
START
LET $SB \leftarrow$ set of secured blocks
LET $MP = []$
FOR i = 1 TO LENGTH(SB)
  x = obtain the integer value of SB[i]
  $x = x \times 16$
  SET solution = FALSE
  WHILE solution = FALSE
    FIND y from the equation $\boldsymbol{Y^2} = (\boldsymbol{x^3} + \boldsymbol{ax}+ \boldsymbol{b})\ \boldsymbol{MOD}$p
    IF y does not have a solution THEN x= x + 1
    ELSE solution = TRUE ENDIF
  ENDWHILE
  MP[i] = (x,y)
ENDFOR
LET mapped points $\leftarrow MP$
STOP
```

Algorithm 8 Encryption of Mapped Points Using Share Node Point (snp)

```
INPUT: mapped points
OUTPUT: encrypted points
START
COMPUTE ssk = $H_s(id_{Ni}) \oplus$ ssk
COMPUTE $snp = ssk \times G$
COMPUTE $snp = snp \times pr_{NiA}$
LET $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \leftarrow$ mapped points
FOR i = 1 TO several mapped points
  $(C_{xi}, C_{yi}) = snp + (x_i, y_i)$
ENDFOR
LET encrypted points $\leftarrow (C_{xi}, C_{yi})$ for $i = 1$ to $n$
STOP
```

including security requirements, resistance of the used cryptography algorithm to security attacks and comparison with the existing models.

A. SECURITY REQUIREMENT

1) CONFIDENTIALITY, INTEGRITY, AND AUTHENTICATION

Each node in the network has a special ID that BS assigns. ECDSA is used for authentication during safe data transmission. Each communication section is secured with unique keys. Broadcast messages are encrypted uniquely for each node on the network. The use of ECDSA in the verification process ensures the message's integrity. Both forward and backward secrecy is maintained in the network as no new node added to the network can decrypt information sent before it joined the network and no node that left the network can decrypt information sent after the node has left the network, hence the scheme also meets the confidentiality criterion. Privacy is preserved in the model as a message sent to a particular node that has to pass through the intermediate node cannot be decrypted by the intermediate node. There is a provision of mutual authentication before establishing communication with any node. Hence, each node is capable of identifying any stranger nodes by checking through nList and ignoring any attempt of a stranger node to establish communication with it.

Algorithm 9 Appending the Signature of the Sending Node to the Encrypted Message (Module Append Signature)

```
INPUT: p, encrypted points
OUTPUT: Signed message
1.  START
2.  LET $C_m \leftarrow$ encrypted points
3.  LET $M_{sent}$=($C_m$, $T$, ) / / Append timestamp T to
    the encrypted message
4.  COMPUTE e = HASH(Msent)
5.  COMPUTE z = le f t most p bits of e
6.  GENERATE random Value k
7.  COMPUTE (x, y) = k × G
8.  COMPUTE c = x mod p where c $\neq$ 0
9.  If c = 0 go to 6
10. COMPUTE d = (z +$pr_{NiA}$∗ c) k −1
11. If d = 0 go to 6
12. ciphertextsignature ← (c,d)
13. STOP
```

Algorithm 10 Verification of Received Signed Message

```
Input: $M_{received}$, $pub_{NiA}$, ciphertextsignature (c,d)
Output: Verified ciphertexts
START
check (c,d) are integers ∈ 1, 2, 3, ..., p − 1)
COMPUTE e = HASH($M_{received}$)
COMPUTE z = leftmost p bits of e
COMPUTE $u1 = e * d^{-1}$ mod p
COMPUTE $u2 = c * d^{-1}$ mod p
COMPUTE (x, y) = u1 ∗ G + u2 ∗$pub_{NiA}$
Verified ← c ≡ x mod p
STOP.
```

2) SCALABILITY, ENERGY CONSERVATION, AND DATA FRESHNESS

Network scalability is the capacity to accommodate network growth. It is sometimes referred to as adding nodes so that network performance is unaffected. The routing techniques for wireless sensor networks should also provide scalability. A decent routing protocol must be scalable and change-adaptive because these routing methods are expected to maintain their performance as the network gets bigger [66]. In the suggested scheme, the adopted clustering technique is more scalable, and energy-efficient and increases the lifetime of WSN [56]. The adoption of hierarchical clustering topology in the scheme ensures that as the network is growing the performance is not degraded. Additional clusters can easily be introduced.

In addition, the localized solutions of clustering ensure dependability and prevent one-point failure. The clustering solution in the proposed WSN model can provide a sleep/wakeup plan that will significantly save power usage.

Algorithm 11 Decryption of Encrypted Points Using Share Node Point (snp)

```
INPUT: encrypted points
OUTPUT: decrypted points
START
COMPUTE ssk = $H_s(id_{NiB}) \oplus$ ssk
COMPUTE $snp = ssk \times G$
COMPUTE $snp = snp \times pub_{NiA} \times pr_{NiB}$
LET $(Cx_1, Cy_1), (Cx_2, Cy_2), \ldots, (Cx_n, Cy_n) \leftarrow$
    $encrypted\ points$
GENERATE array sskseeds[1.n] of seeds using x coordinate of snp
FOR i = 1 TO several encrypted points
$(x_i, y_i) = (Cx_i, Cy_i) - snp$
ENDFOR
LET $decrypted\ points \leftarrow (x_i, y_i)\ for\ i = 1\ to\ n$
STOP
```

Algorithm 12 Decoding Decrypted Points

```
Input: decrypted points, ssk
Output: a set of decoded blocks (in binary digits)
START
COMPUTE ssk = $H_s(id_{NiB}) \oplus$ ssk
COMPUTE $snp = ssk \times G$
COMPUTE $snp = snp \times pub_{NiA}$
LET $\underset{1 \leq i \leq n}{(x_i, y_i)} \leftarrow decrypted\ points$
R = [ ]
FOR i = 1 to n
    X = $x_i \div 16$
    binX = CONVERT X to binary
    S = sskseeds[i]
    L= LENGTH(binX)
    LET pattern = GENERATE array of random
integers(1..L) using S
  as seed
    INITIALIZE msg =[ ]
    FOR col = 1 TO L
      LET msg[pattern[col]] = binX[col]
    ENDFOR
    R[i] = msg
ENDFOR
$set\ of\ decoded\ blocks \leftarrow R$
STOP
```

It is not necessary for all sensor nodes to be awake and use energy in many sensor applications. Some sensor nodes can be put in a sleep mode that uses no energy based on temporal and spatial relationships. These sensor nodes can be given access to an efficient schedule via the BS or CH. Due to its semi-distributed structure, clustering also assures the performance of the application. The use of the multicore wireless sensor in the proposed scheme and the application of the domain decomposition programming model to achieve parallelization of ECC encryption/decryption processes increase the encryption and decryption throughput. Considering equations (7) and (8) the following can be deduced:

$$power = \frac{energy}{time} = Throughput \times force \tag{7}$$

$$energy = power \times time \tag{8}$$

Algorithm 13 Conversion of Binary Values Into Plaintext

```
Input: a set of blocks of binary values
Output: plaintext
LET M = set of blocks of binary values
nblocks = LENGTH(M)
ptext = ‘’ //Empty string
FOR i = 1 TO nblocks
    t = 1
    block = M[i]
    FOR j = 1 TO LENGTH(block)
      B[j] = CONVERT (block [t: t+ 7]) TO ASCII
    value
      B[j] = CONVERT (B[j]) to character
      t = t + 8
    ENDFOR
    $ptext = ptext \parallel B$
ENDFOR
$plaintext \longleftarrow ptext$
STOP
```

(i) the higher the time, the lower the power at constant energy, and (ii) the higher the Throughput at a constant force. From equation (5), the higher the time the higher the energy. Hence, the higher the Throughput the lower the energy. Since the use of multicore sensors as the nodes on the network in the proposed scheme is employed, the encryption/decryption throughput increases which means that the energy consumption of each node reduces. Invariably, the battery life of each node on the network is increased. This means that the battery life of each node will be longer. Data freshness is also guaranteed in the proposed scheme.

Data freshness advocates for sending only current information before any new node is introduced to the network and after any current node exits the network. In the suggested scheme, ssk and snp which are used for encryption and decryption are updated before a node is allowed to join the network and after a current node exits the network. With this measure put in place, a newly added node can only decrypt the ciphertext that it received after joining. The old ciphertext is not accessible to the new nodes. Likewise, a current node that exits the network cannot decrypt the new ciphertext received after exit. Additionally, every sent encrypted message includes a timestamp. As a result, the network is assured of maintaining the freshness of messages exchanged.

B. ROBUSTNESS OF THE PROPOSED SCHEME AGAINST ATTACKS

1) EAVESDROPPING AND SYBIL ATTACKS

Because WSN channels are broadcasting, it is easier for attackers utilizing robust receivers to intercept and eavesdrop on sent data. This data packet capture permits access to a variety of pieces of information, including the position of the nodes, message and node identifiers, timestamps, and application-specific data. In Sybil attacks, malicious nodes in the sensor network offer the other nodes many bogus identifications. There are substantial dangers that could considerably reduce the effectiveness of fault tolerance. In the proposed scheme, authenticated encryption technique is applied. This makes the system to be resistant to both eavesdropping and Sybil attacks [69].

2) MAN-IN-THE-MIDDLE, REPLAY, AND DENIAL OF SERVICE ATTACKS

An attacker in MIMA is capable of impersonating both the sender and receiver [70]. If this attack succeeds, the attacker can send information to the receiver and also give a response to the sender. An attacker in MIMA is also capable of resending the original message sent by a legitimate node to deceive the receiver. The activities of a MIMA attacker can also lead to denial of service attacks if such an attacker decides to regularly transmit false signals to deny authorized network users access to resources or services they are entitled. The proposed scheme is resistant to MIMA attacks of any form. A MIMA attacker that intercepts messages being transmitted between two nodes does not have enough information to compute shared secret keys between the two nodes because it depends on the private keys of both nodes which were not shared. In addition, the MIMA attacker does not have snp which must be manipulated before it is used for encryption/decryption. Because the delivered message contains a timestamp T that identifies the precise instant when the message was sent, a MIMA attack that resolves to engage in replay assaults will fail. This establishes the timing difference used to identify any attack during the replay phase.

A MIMA attacker who resolves to launch denial of service attacks is incapacitated because the attacker needs snp and a shared secret key that has to be computed before it can be used to manipulate snp which is then used to encrypt the data. Access to these two data is only possible through becoming a legitimate user. Any message originating from the MIMA attacker will not be honoured by the legitimate user because the identity of the MIMA attacker cannot be found in nList, the timestamp T will invalidate the message, and discrepancies in randomly generated snp and computation of ssk will make the message invalid.

3) PRIMITIVE ATTACKS

The robustness of the encryption algorithm used in the proposed scheme can be analyzed based on indistinguishability which is normally presented as a game between an authorized user of a cryptosystem and an attacker. If no attacker A, assuming an encrypted version of a message arbitrarily selected from a two-element message space chosen by the attacker, can recognize the message choice with chance considerably better than that of random guessing (1/2), then the cryptosystem is considered secure in terms of distinguishability. Any attacker is deemed to have an advantage in differentiating the ciphertext if they can do so with a probability much higher than 1/2, and the method is not regarded as safe in terms of distinguishability [29]. There are various forms of indistinguishability against different forms of attacks [71] which include: Indistinguishability under chosen plaintext attack (IND-CPA), Indistinguishability under chosen ciphertext attack (IND-CCA), Indistinguishability under non-adaptive chosen ciphertext attack (IND-CCA1), Indistinguishability under adaptive chosen ciphertext attack (IND-CCA2), and Indistinguishability under authenticated adaptive chosen ciphertext attack (IND-CCA3) [72]. While IND-CPA can only provide a guarantee against passive security attacks, IND-CCAs are capable of providing security guarantees against active security attacks [73]. Any cryptography scheme that is evaluated against IND-CCA3 means that it is also verified against IND-CCA2, IND-CCA1, and IND-CPA [29]. Therefore, the encryption scheme used in the proposed WSNs model is verified against IND-CCA3. Formally, the IND-CCA3 advantage measure is defined as follows:

Given Attacker A, and an oracle encryption scheme $\prod = (\boldsymbol{\varkappa}, \boldsymbol{\varepsilon}, \boldsymbol{D})$, the advantage measure of IND-CCA3 is defined by equation (9)

$$Adv_{\prod}^{IND-CCA3}(A) = Pr\left[K \xleftarrow{\$} \varkappa : A^{\varepsilon_K(.),D_K} \Rightarrow 1\right] - Pr\left[A^{\varepsilon_K(\$|.|.),\perp(.)} \Rightarrow 1\right] \quad (9)$$

where $Adv_{\prod}^{IND-CCA3}(A)$ is a measure of the adversarial advantage of IND-CCA3,

Adversary A is a probabilistic algorithm that may have access to the oracle encryption scheme $\prod$, $\prod$ is an oracle encryption scheme with $\boldsymbol{\varkappa}$, $\boldsymbol{\varepsilon and D}$ algorithms, $\boldsymbol{\varkappa}$ is a probabilistic key generation algorithm that returns a key K written as $\boldsymbol{K} \xleftarrow{\$} \boldsymbol{\varkappa}$, $\boldsymbol{\varepsilon}$ is an encryption algorithm that could either be probabilistic or stateful. It takes key K and message M as input to return ciphertext C such that $\boldsymbol{C} \xleftarrow{\$} \boldsymbol{\varepsilon_k(M)}$. If $\boldsymbol{\varepsilon}$ is randomized it flips new coins on each invocation. If it is stateful it uses and then updates the state on each invocation. The oracle $\varepsilon_{\boldsymbol{K}}$ (.), on input M, returns the encryption $\varepsilon_{\boldsymbol{K}}(\boldsymbol{M})$, the oracle $\varepsilon_{\boldsymbol{K}}$ (\$|.|.), on input M, returns the encryption of |M| random bits D is a deterministic decryption algorithm. It takes K and C as input to return M as the output such that $\boldsymbol{M} \xleftarrow{\$} \boldsymbol{D_k(C)}$. The oracle $\boldsymbol{D_k}$ $(\cdot)$, on input C, returns the decryption $\boldsymbol{D_k(C)}$, the oracle $\perp(\cdot)$ returns Invalid on any input. The work [74] showed that all encryption techniques that are IND-CCA3 are also both IND-CPA and AUTH and vice versa. The AUTH notion guarantees the security of ensuring the integrity of ciphertext. Formally, the measure of adversarial advantage of IND-CPA and AUTH are defined by equations (10) and (11) respectively.

$$\mathrm{Adv}_{\prod}^{\mathrm{IND-CPA}}(\mathrm{A}) = \mathrm{Pr}\left[\mathrm{K} \xleftarrow{\$} \varkappa : \mathrm{A}^{\varepsilon_{\mathrm{K}}(.)} \Rightarrow 1\right] - \mathrm{Pr}\left[\mathrm{A}^{\varepsilon_{\mathrm{K}}(\$|.|.)} \Rightarrow 1\right] \quad (10)$$

$$\mathrm{Adv}_{\prod}^{\mathrm{Auth}}(\mathrm{A}) = \mathrm{Pr}\left[\mathrm{K} \xleftarrow{\$} \varkappa : \mathrm{A}^{\varepsilon_{\mathrm{K}}(.)}\mathrm{forges}\right] \quad (11)$$

where A is an adversary that has an encryption oracle ε_{K} (.), Pr[ ] = probability of, and we say that A forges if the encryption oracle ε_{K} (.) outputs a ciphertext C such that C was not the response to any ε_K (M) query and the output of decryption oracle D_K (C) $\neq$ *invalid*.

Equation (11) can be rewritten as an experiment in which the adversary, provided with a ε_K oracle tries to distinguish between a genuine decryption oracle and a fake decryption oracle that returns an Invalid for each input ciphertext. This recast can be represented by equation (12)

$$\mathrm{Adv}_{\prod}^{\mathrm{auth*}}(\mathrm{A}) = \Pr\left[\mathrm{K} \xleftarrow{\$} \varkappa,\ \mathrm{A}^{\varepsilon\mathrm{K}(.),\mathrm{D}_\mathrm{K}(.)} \Rightarrow 1\right] - \Pr\left[\mathrm{A}^{\varepsilon_\mathrm{K}(.),\perp(.)} \Rightarrow 1\right] \quad (12)$$

Equation (**9**) can be rewritten in terms of (**10**) and (**12**) as follows:

$$\begin{aligned}\mathrm{Adv}_{\prod}^{\mathrm{IND-CCA3}}(\mathrm{A}) &= \Pr\left[\mathrm{K} \xleftarrow{\$} \varkappa : \mathrm{A}^{\varepsilon_\mathrm{K}(.),\mathrm{D}_\mathrm{K}} \Rightarrow 1\right] \\ &\quad - \Pr\left[\mathrm{A}^{\varepsilon_\mathrm{K}(\$|.|.),\perp(.)} \Rightarrow 1\right] \\ &= \left(\Pr\left[\mathrm{K} \xleftarrow{\$} : \mathrm{A}^{\varepsilon_\mathrm{K}(.),\mathrm{D}_\mathrm{K}(.)} \Rightarrow 1\right]\right. \\ &\quad \left. - \Pr\left[\mathrm{A}^{\varepsilon_\mathrm{K}(.),\perp(.)} \Rightarrow 1\right]\right) \\ &\quad + \left(\Pr\left[\mathrm{A}^{\varepsilon_\mathrm{K}(.),\perp(.)} \Rightarrow 1\right]\right. \\ &\quad \left. - \Pr\left[\mathrm{A}^{\varepsilon_\mathrm{K}(\$|.|.),\perp(.)} \Rightarrow 1\right]\right) \quad (13)\end{aligned}$$

Let B1 and B2 be the two adversaries capable of IND-CPA and AUTH, respectively, then from equation (13), the measure of adversarial advantage $Adv_{\prod}^{IND-CCA3}(A)$ can be measured in terms of the adversarial advantage of B1 and B2 as depicted in equation 14.

$$Adv_{\prod}^{IND-CCA3}(A) = Adv_{\prod}^{IND-CPA}(B1) + Adv_{\prod}^{auth*}(B2) \quad (14)$$

A cryptosystem is said to be secure under $IND-CCA3$ if the adversarial advantage $Adv_{\prod}^{IND-CCA3}(A)$ is negligible.

Proof: There are two portions to the proof of equation (14). First, the proof that establishes that the adversarial advantage of B1($Adv_{\prod}^{IND-CPA}(B1)$) is negligible. A scheme is indistinguishable under $IND-CPA$ if the adversary has the power to submit any $m_{i,0}m_{i,1}$ i = [1,2,..,q] encryption queries as many as desired and receives the ciphertext $c_i \leftarrow E(k, m_i, b)$ for each message. Where $E(k, m_i, b)$ represents encryption of the message m_i, b scheme using k as the encryption key and b = {0,1}. The adversary cannot modify b and is not aware of the value of b that is passed to the encryption oracle and is limited to a polynomial running time. The adversary may guess and under this condition, he/she will guess right in about half of all the cases. To prove that $Adv_{\prod}^{IND-CPA}(B1) = negligible$, the definition of $IND-CPA$ which is written as: $Adv_{\prod}^{IND-CPA}(B1, E) = |pr[EXP(0) = 1] - pr[EXP(1) = 1]|$. In this definition, adversary B1 is capable of accessing encryption oracle E and performing two experiments EXP(0) and EXP(1). In EXP(0), two messages of the same length ($m_{i,0}$ and $m_{i,1}$) are submitted to E by the attacker and the output $c_{i,b} \leftarrow E(k, m_i, b)$ is obtained. In EXP(1), the attacker may enter either ($m_{i,0}$ and $m_{i,0}$) or ($m_{i,1}$ and $m_{i,1}$) and the encryption oracle outputs $c_{i,b}$ ← E(k, $m_{i,b}$) is obtained. The attacker can then contrast the findings and ascertain the worth of b = 0, 1. Figure 4 illustrates the attacker's steps to perform CPA.

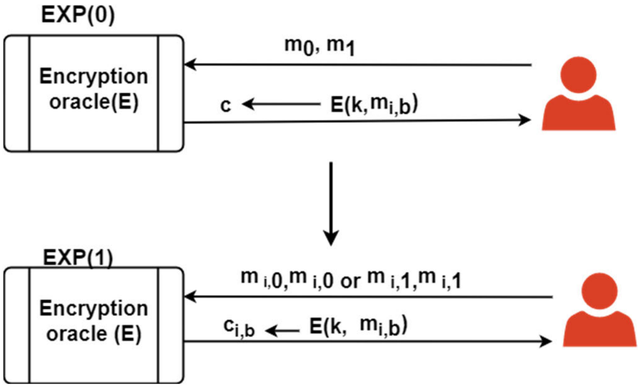

FIGURE 4. Attackers' steps to perform CPA.

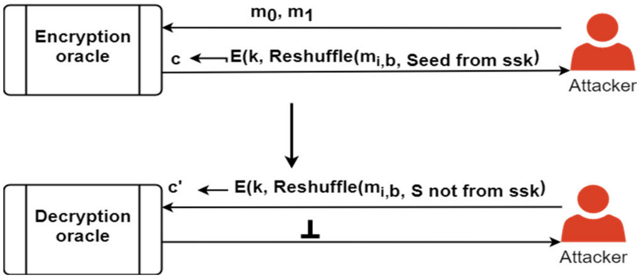
Encryption oracle
m0, m1
c
E(k, Reshuffle(mi,b, Seed from ssk)
Attacker
Decryption oracle
c'
E(k, Reshuffle(mi,b, S not from ssk)
⊥
Attacker

FIGURE 5. Resistance of the propose encryption scheme to auth attack.

However, in the encryption scheme used in the proposed WSNs model, random seeds generated from the x coordinate of snp are used for reshuffling the bits of $m_{i,b}$ in each encryption session (experiment), $c_{i,b}$ $\leftarrow <$ $E(k, reshuffle(m_{i,b}, seedi)$ $>$. Therefore, each time the attacker inputs the same messages ($m_{i,j}$ and $m_{i,j}$) the encryption oracle gives $c_{i,j} \neq c_{i+1,j}$ as output. Hence, $\mathrm{Adv}_{\prod}^{\mathrm{IND-CPA}}(\mathrm{B1})$ = negligible.

To prove that $Adv_{\prod}^{auth*}(B2)$ is negligible, it should be noted that adversary B2 is capable of access to both encryption and decryption oracles where the c_i presented to the decryption oracle is not equal to the c_i received from encryption oracle. B2 can modify the received c_i by reshuffling its bits using a random seed s not generated from ssk to guess m_i. However, as authentication of the ciphertext is carried out before decryption, the decryption oracle in the proposed model ignores any modified $\mathrm{c_i}$, meaning that the oracle returns $\perp \leftarrow <$ D(k, c'_i,) $>$ for each modified c'_i,. Figure 5 illustrates how the proposed encryption model is capable of resisting the auth attack. As a result, the $Adv_{\prod}^{auth*}(B2) = negligible$.

Since $\mathrm{Adv}_{\prod}^{\mathrm{IND-CPA}}(\mathrm{B1}) = Adv_{\prod}^{auth*}(B2) = negligible$, from equation 11, $Adv_{\prod}^{IND-CCA3}(A) = Adv_{\prod}^{IND-CPA}(B1) + Adv_{\prod}^{auth*}(B2) = negligible + negligible = negligible$.

TABLE 2. Comparative analysis of the proposed and existing WSN models.

WSNs Models	Cryptography Method	1	2	3	4	5	6
Proposed Model	ECC, ECDH & ECDSA	✓	✓	✓	✓	✓	✓
[6]	DH & Knapsack Algorithm	✓	✓	×	✓	✓	×
[44]	Modified ECC & digital hashing	✓	✓	×	✓	✓	×
[41]	ECDH RSA	✓	✓	×	✓	✓	×
[64]	ECDSA	×	×	✓	✓	×	×
[75]	ECC	×	✓	✓	×	×	×
[76]	Encryption and Exclusive OR	×	✓	✓	✓	✓	×

C. COMPARATIVE ANALYSIS OF THE PROPOSED WSNs MODEL AND EXISTING MODELS

The following parameters were used to compare the proposed WSN to the ones already in existence: 1: Confidentiality, 2: Integrity, 3: Authentication, 4: Scalability, 5: Data freshness 6: support for multiprocessing as comparative metrics. Table 2 displays the results of the comparison of the proposed model with some existing models in terms of functionalities.

D. COMPARATIVE PERFORMANCE ANALYSIS OF OPTIMIZED ECC USED IN THE PROPOSED SCHEME

The essence of introducing a multicore processor and ECC-based encryption/decryption scheme that adopts a domain decomposition programming model to achieve parallelization is to increase the speed of processing and thereby reduce the energy consumption of each node during encryption and decryption processes. It is therefore pertinent to verify if this aim is achieved. To verify whether the approach leads to a reduction in power consumption or not, data of different sizes were encrypted using the ECC encryption scheme in [33] and the proposed encryption scheme on uniprocessor and dual-core systems. Experiments were set up to measure the encryption time and decryption time of data.

Different data sizes (in bytes) were used. When a particular key is generated, the encryption and decryption processes are carried out in 10 rounds each. In each round, the time taken for the encryption/decryption to complete was measured. These 10 different execution times were added together and the average time was taken as the encryption/decryption time. This procedure was used for each of the two schemes that were used for the performance comparison on both single and dual-core laptops. Encryption/decryption throughput and energy consumption during encryption/decryption processes were determined.

E. COMPARATIVE PERFORMANCE ANALYSIS BASED ON ENCRYPTION/DECRYPTION THROUGHPUT

Throughput was determined using the formulae in equations (15) and (16).

$$Encryption\ Throughput = \frac{Data\ size}{Time\ taken\ to\ encrypt\ the\ data} \tag{15}$$

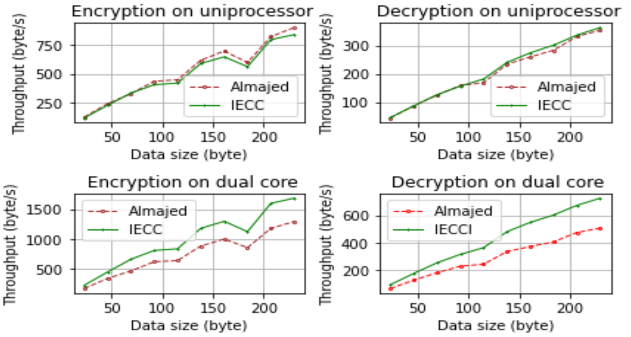

FIGURE 6. Comparison of encryption and decryption throughputs of the ECC scheme of Almajed et. al, and the ECC scheme proposed in this research work.

$$Decryption\ Throughput = \frac{Data\ size}{Time\ taken\ to\ decrypt\ the\ data} \tag{16}$$

The graphs in Figure 6 show the encryption and decryption throughputs when the ECC scheme proposed by Almajed and the proposed ECC in this research work is tested on both a uniprocessor system and a dual-core system.

Figure 6 shows that the two schemes behave similarly in terms of encryption/decryption throughput on a uniprocessor system. However, the performance of IECC, the ECC cryptosystem proposed in this research has higher throughputs for both encryption and decryption procedures on the dual-core system. These results show that the proposed scheme if used in WSN, will improve the performance of the WSN.

F. COMPARATIVE PERFORMANCE ANALYSIS BASED ON ENCRYPTION/DECRYPTION ENERGY CONSUMPTION

The approximate average current consumption of the laptop used for the experiment while it is busy and the CPU voltage were both acquired from the manual of the HP laptops used in the experiments to calculate the energy consumption by the system during encryption/decryption processes. The average current consumption I on the uniprocessor system that was used is 100mA, and the CPU voltage V_{cc} is 1.25v while on the dual-core, the current is 120mA and CPU voltage is 1.3v. The Energy consumption E in Joule (J) of the laptops used for the experiments during encryption/decryption was calculated using the formula: $E = V_{cc} \times I \times t$, where t represents encryption/ decryption time. The graphs shown in Figure 6 show the results obtained.

From Figure 7, the uniprocessor system consumes a similar quantity of energy during encryption and decryption when Almajed and the proposed cryptosystems (IECC) are used. On the other hand, there is a drop in energy consumption by the dual-core system when the two cryptosystems are run. However, there is a significant drop in energy consumption when the proposed cryptosystem is run on the dual-core system. It can, therefore, be said that the use of the multicore processor, will reduce energy consumption when it is used in

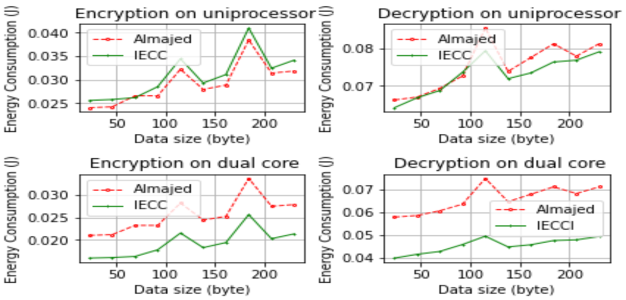

FIGURE 7. Energy consumption during encryption and decryption on uniprocessor and dual core processor.

multicore WSNs alongside the proposed IECC cryptosystem in this research work.

G. PERFORMANCE ANALYSIS BASED ON ENHANCEMENT TO MAPPING SCHEME

When a block of plaintext to be encrypted is converted to an integer value which is normally represented as an x coordinate value on an elliptic curve. It is necessary to find the value of the y coordinate to create a point (x, y) on the elliptic curve. Usually, the y coordinate does not fit the EC's equation. Consequently, it is impossible to transfer the x coordinate of (x, y) locations to the EC. Therefore, until a value is found that matches, x must be increased by 1 and y must be recalculated. The point (x, y) is ultimately mapped as a result. This is done to all the integer values that represent the blocks in the plaintext to be encrypted. However, the steps explained above change the original value of x and by implication, the original plaintext. Two techniques in the literature are normally used to overcome this problem: the probability method introduced by [77] and appending method.

The value k which denotes the number of rounds necessary to map the integer to the point on EC is determined using the probability strategy. This value is multiplied by x during the mapping step, with the possibility of additional k rounds. This value of x can be obtained by determining the value of the transferred point and applying the formula: $\left\lfloor \frac{\boldsymbol{mappedpoint}}{\boldsymbol{k}} \right\rfloor$. In the adding procedure, x has n bits appended to it. When n bits are appended, the maximum amount of rounds that can be used in the mapping phase without risk is $\boldsymbol{2^n}$ rounds. The ability to give a maximum number of rounds based on the attached bits is an advantage of the adding approach, but the complexity of the algorithm for concatenating two texts is $\boldsymbol{O(n^2)}$ [78], where n is the bit size of the numerical value. The complexity of multiplication operation using [79] is $\boldsymbol{O(nlog4^{log*}n)}$. The comparison of the complexities of concatenation and multiplication operations can be pictorially illustrated. Complexity comparison of concatenation and multiplication operation would reflect a steady growth rate.

The fact that the multiplication operation is less in complexity than the concatenation operation implies that it is less hard to use the probability approach than the adding method. However, the probability technique has the disadvantage of only supporting a smaller number of round increments than the adding method [33]. In this research, to improve performance, the probability method is adopted. Additionally, the appending method is employed to improve the effectiveness of the choice of k. This was achieved by recalculating the number of rounds (n round) from k using the formula: $\boldsymbol{nround} = \boldsymbol{2^k}$. The effect of this approach is checked by setting up an experiment where the values of k were varied to determine the maximum number of rounds for both enhanced probability methods in the proposed method and that of [33]. As can be seen from the values in Table 3 and their representation by the clustered chart in Figure 8, the enhanced probability employed in the proposed scheme allows for more rounds than that of Almajed and Almongren [33].

TABLE 3. Comparative analysis of the proposed enhanced probability method in Almajed and Almogren [33].

Experiment	k	[33] $2^{\lceil \log_2 k \rceil}$	Proposed scheme 2^k
1	2	2	4
2	4	4	16
3	6	8	64
4	8	8	256
5	10	16	1024
6	12	16	4096

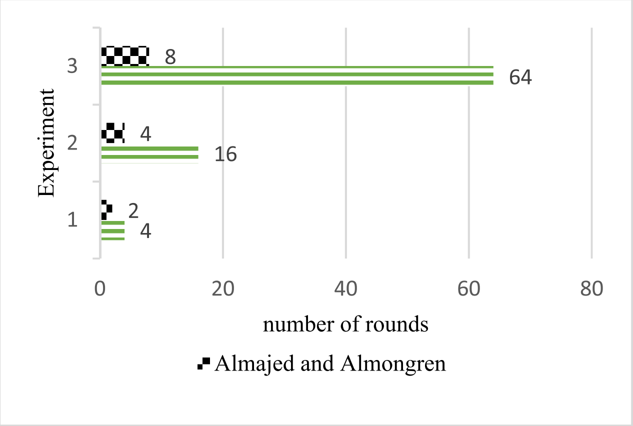

FIGURE 8. Enhanced probability techniques in [33] versus the proposed scheme.

V. CONCLUSION AND FUTURE SCOPE

This paper focused on improving the security of WSNs and reducing the energy consumption of the sensors to ensure the longevity of their life span. Researchers have employed the clustering of the nodes in the WSNs model as a measure to reduce power consumption in the network. However, researchers have not explored the possibility of using multicore WS as nodes on WSNs models. In this paper, a WSN model that combines node clustering techniques and multicore wireless sensors to effect a further reduction in the power consumption of WSNs is proposed. The existing WSN model was optimized to resist communication with stranger nodes by introducing mutual authentication between the communicating nodes. In the same vein, an optimized ECC algorithm that employs the use of ECDH, ECDSA, was employed to ensure the CIA of the model. The optimized ECC algorithm uses a domain decomposition programming model, enabling efficient resource use in multicore/multiprocessing instances. A comparative analysis of the proposed model with the existing model was carried out, and the results reveal that the proposed model maintains the security of WSNs and can conserve energy more than existing models. Hence, the proposed WSNs model can be employed in various applications where security and reduction in power consumption are required. Future work would focus on designing and implementing optimal power consumption of multicore WSN systems.

ABBREVIATIONS

BS	Base Station.
CPUs	Central Processing Units.
CCRM	Chessboard Clustering Routing Method.
CCA	Chosen Ciphertext Attack.
CPA	Chosen Plaintext Attack.
CBC	Cipher Block Chaining.
CH	Cluster Head.
CIA	Confidentiality, Integrity, and Authenticity.
ECC	Elliptic Curve Cryptography.
ECDH	Elliptic Curve Diffie-Helman.
ECDSA	Elliptic Curve Digital Signature Algorithm.
ECPM	Elliptic Curve Point Multiplier.
ECSM	Elliptic Curve Scalar Multiplication.
EAFCA	Energy Aware Fuzzy Clustering Algorithm.
GPS	Global Positioning System.
IDN-CPA	Indistinguishability Under Chosen Plaintext Attack.
IND-CCA	Indistinguishability Under Chosen-Ciphertext Attack.
IC	Integrated Circuit.
KPA	Known plaintext Attack.
MiMA	Man-in-the-Middle-Attack.
MIMD	Multiple Instruction Multiple Data.
MISD	Multiple Instruction Single Data.
NUMA	Non-Uniform Memory Access.
RRE	Remaining Residual Energy.
SIMD	Single Instruction Multiple Data.
SISD	Single Instruction Single Data.
SMPs	Symmetric Multi-Processings.
TCHs	Tentative Cluster Heads.
UMA	Uniform Memory Access.
WMSN	Wireless Multimedia Sensor Networks.
WSNs	Wireless Sensors Networks.
WS	Wireless Sensors.

ACKNOWLEDGMENT

The authors would like to thank the Editor-in-Chief and the referees for many valuable comments and suggestions which have resulted in several improvements to the presentation of the paper.

DATA AVAILABILITY STATEMENT

The data related to the outcome of this study are available upon reasonable request from the first author.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] M. Matin and M. Islam, “Overview of wireless sensor network,” in *Wireless Sensor Networks—Technology and Protocols*. London, U.K.: IntechOpen, 2012. [Online]. Available: https://www.intechopen.com/chapters/38793, doi: 10.5772/49376.

[2] M. P. Durisic, “A survey of military applications of wireless sensor networks,” in *Proc. Medit. Conf. Embedded Comput. (MECO)*, Jun. 2012, pp. 38–74.

[3] A. Kumar, B. S. Dhaliwal, and D. Singh, “Energy efficient clustering protocols for wireless sensor networks: A review,” *Webology*, vol. 18, no. 4, pp. 391–404, 2021. [Online]. Available: http://www.webology.org

[4] O. J. Pandey and R. M. Hegde, “Low-latency and energy-balanced data transmission over cognitive small world WSN,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7719–7733, Aug. 2018.

[5] M. R. Senouci and A. Mellouk, “A robust uncertainty-aware cluster-based deployment approach for WSNs: Coverage, connectivity, and lifespan,” *J. Netw. Comput. Appl.*, vol. 146, Nov. 2019, Art. no. 102414.

[6] K. A. Ameen, B. A. Mahmood, and Y. N. A. Taher, “Secure message transmission scheme in wireless sensor networks,” *Bull. Electr. Eng. Informat.*, vol. 10, no. 3, pp. 1514–1523, Jun. 2021, doi: 10.11591/EEI.V10I3.2856.

[7] K. Viji, T. S. Perumal, R. S. Prakash, and M. V. Ananthkumar, “An improved three-layer low-energy adaptive clustering hierarchy for wireless sensor networks,” *Int. J. Innov. Sci. Res. Technol.*, vol. 2, no. 5, pp. 797–803, 2017, doi: 10.1109/JIOT.2016.2530682.

[8] A. Rani and S. Kumar, “A survey of security in wireless sensor networks,” in *Proc. 3rd Int. Conf. Comput. Intell. Commun. Technol. (CICT)*, Feb. 2017, pp. 1–5, doi: 10.1109/CIACT.2017.7977334.

[9] T. A. Alghamdi, “Energy efficient protocol in wireless sensor network: Optimized cluster head selection model,” *Telecommun. Syst.*, vol. 2020, pp. 1–15, Mar. 2020, doi: 10.1007/S11235-020-00659-9.

[10] *National Diploma in Computer Technology Introduction to Computing*, UNESCO-TVE, Abuja, Nigeria, 2008, pp. 1–81.

[11] R. G. Vieira, A. M. da Cunha, L. B. Ruiz, and A. P. de Camargo, “On the design of a long range WSN for precision irrigation,” *IEEE Sensors J.*, vol. 18, no. 2, pp. 773–780, Jan. 2018.

[12] T. Hintsch and S. Irnich, “Large multiple neighborhood search for the clustered vehicle-routing problem,” *Eur. J. Oper. Res.*, vol. 270, no. 1, pp. 118–131, 2018.

[13] A. Ahmad, N. Javaid, Z. A. Khan, U. Qasim, and T. A. Alghamdi, “$(ACH)^2$: Routing scheme to maximize lifetime and through-put of wireless sensor networks,” *IEEE Sensors J.*, vol. 14, no. 10, pp. 3516–3532, Oct. 2014.

[14] T. A. Alghamdi, “Cluster based energy efficient routing protocol for wireless body area networks,” *Trends Appl. Sci. Res.*, vol. 11, no. 1, pp. 12–16, 2016.

[15] T. A. Alghamdi, “Secure and energy efficient path optimization technique in wireless sensor networks using DH method,” *IEEE Access*, vol. 6, pp. 53576–53582, 2018.

[16] M. Krishnan, S. Yun, and Y. M. Jung, “Enhanced clustering and ACO-based multiple mobile sinks for efficiency improvement of wireless sensor networks,” *Comput. Netw.*, vol. 160, pp. 33–40, Sep. 2019.

[17] S. Radhika and P. Rangarajan, “On improving the lifespan of wireless sensor networks with fuzzy based clustering and machine learning based data reduction,” *Appl. Soft Comput.*, vol. 83, Oct. 2019, Art. no. 105610.

[18] S. A. Jesudurai and A. S. Kumar, “An improved energy efficient cluster head selection protocol using the double cluster heads and data fusion methods for IoT applications,” *Cogn. Syst. Res.*, vol. 57, pp. 101–106, Oct. 2019.

[19] B. Zhao, Y. Ren, D. Gao, L. Xu, and Y. Zhang, “Energy utilization efficiency evaluation model of refining unit based on contourlet neural network optimized by improved grey optimization algorithm,” *Energy*, vol. 185, pp. 1032–1044, Oct. 2019.

[20] J. Goodacre. (2022). *The Design Dilemma_Multiprocessing Using Multiprocessors and Multithreading*. Accessed: Apr. 11, 2022. [Online]. Available: https://www.design-reuse.com/contact/

[21] N. N. Sirhan and S. I. Serhan, “Multi-core processors: Concepts and implementations,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 10, no. 1, pp. 1–10, Feb. 2018, doi: 10.5121/IJCSIT.2018.10101.

[22] P. Kastnes. (2020). *Power Consumption Explained*. Accessed: Oct. 10, 2022. [Online]. Available: https://blog.nordicsemi.com/getconnected/power-consumption-explained#:~:text=Power consumptionexplained1Powerconsumptionisoften,digitalgates..4Areal-worldexample

[23] K. Castro. (2018). *Multiprocessor Systems.pdf*. [Online]. Available: https://www.tutorialspoint.com/index.htm

[24] D. M. Schinianakis, A. P. Fournaris, A. P. Kakarountas, and T. Stouraitis, "An RNS architecture of an F_p elliptic curve point multiplier," in *Proc. ISCAS*, May 2006, pp. 3359–3373, doi: 10.1109/ISCAS.2006.1693348.

[25] K. Jarvinen and J. Skytta, "On parallelization of high-speed processors for elliptic curve cryptography," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 16, no. 9, pp. 1162–1175, Sep. 2008, doi: 10.1109/TVLSI.2008.2000728.

[26] Y. Zhang, D. Chen, Y. Choi, L. Chen, and S. Ko, "A high performance pseudo-multi-core ECC processor over GF(2^{163})," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2010, pp. 701–704, doi: 10.1109/ISCAS.2010.5537486.

[27] A. Bellemou, N. Benblidia, M. Anane, and M. Issad, "MicroBlaze-based multiprocessor embedded cryptosystem on FPGA for elliptic curve scalar multiplication over F_p," *J. Circuits, Syst. Comput.*, vol. 28, no. 3, Mar. 2019, Art. no. 1950037, doi: 10.1142/S0218126619500373.

[28] K. V. Gurudutt, "Considerations in software design for multicore multiprocessor architectures," IBM, Armonk, NY, USA, Tech. Rep., Aug. 2019. [Online]. Available: https://developer.ibm.com/articles/au-aix-multicore-multiprocessor/

[29] H. N. Almajed and A. S. Almogren, "SE-ENC: A secure and efficient encoding scheme using elliptic curve cryptography," *IEEE Access*, vol. 7, pp. 175865–175878, 2019.

[30] A. Sengupta and U. K. Ray, "Message mapping and reverse mapping in elliptic curve cryptosystem," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 5363–5375, 2016, doi: 10.1002/SEC.1702.

[31] J. Muthukuru and B. Sathyanarayana, "Fixed and variable size text based message mapping techniques using ECC," *Global J. Comput. Sci. Technol.*, vol. 12, no. 3, pp. 12–18, 2012.

[32] F. Amounas and E. E. Kinani, "Fast mapping method based on matrix approach for elliptic curve cryptography," *Int. J. Inf. Netw. Secur.*, vol. 1, no. 2, pp. 54–59, 2012.

[33] H. N. Almajed and A. S. Almogren, "A secure and efficient ECC-based scheme for edge computing and Internet of Things," *Sensor*, vol. 20, no. 21, pp. 1–31, 2020, doi: 10.3390/S20216158.

[34] E. T. Oladipupo and O. C. Abikoye, "Improved authenticated elliptic curve cryptography scheme for resource starve applications," *Comput. Sci. Inf. Technol.*, vol. 3, no. 3, pp. 169–185, 2022.

[35] K. Keerthi and B. Surendiran, "Elliptic curve cryptography for secured text encryption," in *Proc. Int. Conf. Circuit, Power Comput. Technol. (ICCPCT)*, Apr. 2017, pp. 1–5.

[36] Y. Yin, L. Wu, Q. Peng, and X. Zhang, "A novel SPA on ECC with modular subtraction," in *Proc. 12th IEEE Int. Conf. Anti-Counterfeiting, Secur., Identificat. (ASID)*, Nov. 2018, pp. 179–182.

[37] S. D. Galbraith and F. Vercauteren, "Computational problems in supersingular elliptic curve isogenies," *Quantum Inf. Process.*, vol. 17, no. 10, p. 265, Oct. 2018.

[38] T. Shahroodi, S. Bayat-Sarmadi, and H. Mosanaei-Boorani, "Low-latency double point multiplication architecture using differential addition chain over $GF(2^m)$," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 4, pp. 1465–1473, Apr. 2019.

[39] S. J. Bigelow. *Definition: Multicore Processor*. Accessed: Sep. 16, 2022. [Online]. Available: https://www.techtarget.com/searchdatacenter/definition/multi-core-processor

[40] L. Null and J. Lobur, *The Essentials of Computer Organization and Architecture*. Burlington, MA, USA: Jones and Bartlett, 2003.

[41] B. Abood, A. N. Faisal, and Q. A. Hamed, "Data transmitted encryption for clustering protocol in heterogeneous wireless sensor networks," *Indonesian J. Elect. Eng. Comput. Sci.*, vol. 25, no. 1, pp. 347–357, 2022, doi: 10.11591/IJEECS.V25.I1.PP347-357.

[42] X. Zhang and J. Wang, "An efficient key management scheme in hierarchical wireless sensor networks," in *Proc. Int. Conf. Comput., Commun. Secur. (ICCCS)*, Dec. 2015, pp. 1–7, doi: 10.1109/CCCS.2015.7374122.

[43] K. Chatterjee, A. De, and D. Gupta, "A secure and efficient authentication protocol in wireless sensor network," *Wireless Pers. Commun.*, vol. 81, no. 1, pp. 17–37, Mar. 2015, doi: 10.1007/s11277-014-2115-2.

[44] B. Patil and S. R. Biradar, "An efficient authentication and key-distribution protocol for wireless multimedia sensor network," *Indonesian J. Elect. Eng. Comput. Sci.*, vol. 27, no. 1, pp. 347–354, 2022, doi: 10.11591/ijeecs.v27.i1.pp347-354.

[45] J. Zhang, Q. Cui, and X. Liu, "An efficient key management scheme for wireless sensor networks in hostile environments," in *Proc. Int. Conf. Multimedia Inf. Netw. Secur.*, 2009, pp. 417–420, doi: 10.1109/MINES.2009.157.

[46] S. K. Gupta, N. Jain, and P. Sinha, "Clustering protocols in wireless sensor networks: A survey," *Int. J. Appl. Inf. Syst.*, vol. 5, no. 2, pp. 41–50, 2013. [Online]. Available: http://www.ijais.org

[47] S. Bandyopadhyay and E. J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *Proc. 20th Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM)*, vol. 3, Apr. 2003, pp. 1713–1723.

[48] S. Parvin, S. Han, Z. U. Rehman, A. Al Faruque, and F. K. Hussain, "A new identity-based group signature scheme based on knapsack ECC," in *Proc. 6th Int. Conf. Innov. Mobile Internet Services Ubiquitous Comput.*, Jul. 2012, pp. 73–80, doi: 10.1109/IMIS.2012.88.

[49] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. Mobile Comput.*, vol. 4, no. 4, pp. 366–379, Oct./Dec. 2004, doi: 10.1109/TMC.2004.41.

[50] A. D. Amis, R. Prakash, T. H. P. Vuong, and D. T. Huynh, "Max-min d-cluster formation in wireless ad hoc networks," in *Proc. Conf. Comput. Commun., 19th Annu. Joint Conf. IEEE Comput. Commun.*, Mar. 2000, pp. 32–41, doi: 10.1109/INFCOM.2000.832171.

[51] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Annu. Hawaii Int. Conf. Syst. Sci.*, Jan. 2000, pp. 1–10, doi: 10.1109/HICSS.2000.926982.

[52] T. Ahmad, M. Haque, and A. M. Khan, "An energy-efficient cluster head selection using artificial bees colony optimization for wireless sensor networks," in *Advances in Nature-Inspired Computing and Applications* (EAI/Springer Innovations in Communication and Computing), S. K. Shandilya, S. Shandilya, A. K. Nagar, Eds. Cham, Switzerland: Springer, 2019, pp. 189–203.

[53] E. Alnawafa and I. Marghescu, "New energy efficient multi-hop routing techniques for wireless sensor networks: Static and dynamic techniques," *Sensors*, vol. 18, no. 6, p. 1863, Jun. 2018, doi: 10.3390/s18061863.

[54] S. B. Kamble and V. V. Jog, "Efficient key management for dynamic wireless sensor network," in *Proc. 2nd IEEE Int. Conf. Recent Trends Electron., Inf. Commun. Technol. (RTEICT)*, May 2017, pp. 583–586, doi: 10.1109/RTEICT.2017.8256663.

[55] P. Kathiroli and K. Selvadurai, "Energy efficient cluster head selection using improved sparrow search algorithm in wireless sensor networks," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 10, pp. 8564–8575, Nov. 2022, doi: 10.1016/j.jksuci.2021.08.031.

[56] I. S. Akila, S. V. Manisekaran, and R. Venkatesan, "Modern clustering techniques in wireless sensor networks," in *Wireless Sensor Networks-Insights and Innovations*, P. J. Sallis, Ed. Rejika, Croatia: INTECH, 2017, pp. 141–156.

[57] M. M. Raouf, "Clustering in wireless sensor networks (WSNs)," *J. Baghdad Univ. College Econ. Sci.*, vol. 57, pp. 1–9, Mar. 2019, doi: 10.13140/RG.2.2.34342.98887.

[58] H. Rhim, K. Tamine, R. Abassi, D. Sauveron, and S. Guemara, "A multi-hop graph-based approach for an energy-efficient routing protocol in wireless sensor networks," *Hum.-Centric Comput. Inf. Sci.*, vol. 8, no. 1, pp. 1–21, Dec. 2018, doi: 10.1186/s13673-018-0153-6.

[59] J. Sumathi and R. L. Velusamy, "A review on distributed cluster based routing approaches in mobile wireless sensor networks," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 1, pp. 1–15, 2020, doi: 10.1007/s12652-020-02088-7.

[60] M. Khodiaeva, S. Obeidat, D. Salane, and J. Holst, "Security architecture framework for Internet of Things (IoT)," in *Proc. IEEE 10th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf.*, Oct. 2020, pp. 154–157.

[61] K. K. F. Yuen, "Towards a cybersecurity investment assessment method using primitive cognitive network process," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Feb. 2019, pp. 68–71.

[62] C. Biswas, U. D. Gupta, and M. M. Haque, "An efficient algorithm for confidentiality, integrity and authentication using hybrid cryptography and steganography," in *Proc. Int. Conf. Electr., Comput. Commun. Eng. (ECCE)*, Cox'sBazar, Bangladesh, Feb. 2019, pp. 1–5.

[63] R. T. Tiburski, C. R. Moratelli, S. F. Johann, M. V. Neves, E. de Matos, L. A. Amaral, and F. Hessel, "Lightweight security architecture based on embedded virtualization and trust mechanisms for IoT edge devices," *IEEE Commun. Mag.*, vol. 57, no. 2, pp. 67–73, Feb. 2019.

[64] W.-H. Wang, Y.-L. Cui, and T.-M. Chen, "Design and implementation of an ECDSA-based identity authentication protocol on WSN," in *Proc. 3rd IEEE Int. Symp. Microw., Antenna, Propag. EMC Technol. Wireless Commun.*, Oct. 2009, pp. 1202–1205, doi: 10.1109/MAPE.2009.5355821.

[65] P. Sinha, V. K. Jha, A. K. Rai, and B. Bhushan, "Security vulnerabilities, attacks and countermeasures in wireless sensor networks at various layers of OSI reference model: A survey," in *Proc. Int. Conf. Signal Process. Commun. (ICSPC)*, Jul. 2017, pp. 288–293, doi: 10.1109/CSPC.2017.8305855.

[66] R. Priyadarshi, B. Gupta, and A. Anurag, "Deployment techniques in wireless sensor networks: A survey, classification, challenges, and future research issues," *J. Supercomput.*, vol. 76, pp. 7333–7373, Jan. 2020, doi: 10.1007/s11227-020-03166-5.

[67] B. A. Mahmood and D. Manivannan, "Position based and hybrid routing protocols for mobile ad hoc networks: A survey," *Wireless Pers. Commun.*, vol. 83, no. 2, pp. 1009–1033, Jul. 2015.

[68] B. Mahmood, A. Ibrahim, and D. Manivannan, "SAriadne: A secure source routing protocol to prevent hidden-channel attacks," in *Proc. IEEE 12th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2016, pp. 1–7, doi: 10.1109/WiMOB.2016.7763267.

[69] V. E. Ekong and U. O. Ekong, "A survey of security vulnerabilities in wireless sensor networks," *Nigerian J. Technol.*, vol. 35, no. 2, pp. 392–397, 2016, doi: 10.4314/njt.v35i2.21.

[70] S. Gupta, H. K. Verma, and A. L. Sangal, "Security attacks & prerequisite for wireless sensor networks," *Int. J. Eng. Adv. Technol.*, vol. 2, no. 5, pp. 558–566, 2013.

[71] S. Debnath, M. V. Nunsanga, and B. Bhuyan, "Study and scope of signcryption for cloud data access control," in *Advances in Computer, Communication and Control*. Singapore: Springer, 2019, pp. 113–126.

[72] C. Boyd, B. Hale, S. F. Mjølsnes, and D. Stebila, "From stateless to stateful: Generic authentication and authenticated encryption constructions with application to TLS," in *Proc. Cryptographers' Track RSA Conf.*, 2016, pp. 55–71.

[73] M. Abdalla, F. Benhamouda, and D. Pointcheval, "Public-key encryption indistinguishable under plaintext-checkable attacks," *IET Inf. Secur.*, vol. 10, no. 6, pp. 288–303, Nov. 2016, doi: 10.1049/iet-ifs.2015.0500.

[74] T. Shrimpton, "A characterization of authenticated-encryption as a form of chosen-ciphertext security," Cryptol. ePrint Arch., Paper 2004/272, vol. 2004, no. 272, 2004, pp. 1–7. [Online]. Available: https://eprint.iacr.org/2004/272

[75] R. Maharana and P. M. Khilar, "An improved authentication protocol for hierarchical wireless sensor networks using ECC," *Int. J. Comput. Appl.*, vol. 67, no. 2, pp. 23–30, 2013.

[76] O. Cheikhrouhou, A. Koubaa, M. Boujelben, and M. Abid, "A lightweight user authentication scheme for wireless sensor networks," in *Proc. ACS/IEEE Int. Conf. Comput. Syst. Appl.*, May 2010, pp. 1–7, doi: 10.1109/AICCSA.2010.5586995.

[77] B. King, "Mapping an arbitrary message to an elliptic curve when defined over $GF(2^n)$," *Int. J. Netw. Secur.*, vol. 8, no. 2, pp. 169–176, 2009.

[78] B. Rahman, "We don't need StringBuilder for simple concatenation-DZone Java," DZone, Cary, NC, USA, Tech. Rep., 2019. [Online]. Available: https://dzone.com/articles/string-concatenation-performacne-improvement-in-ja

[79] D. Harvey and J. Van Der Hoeven, "Faster integer multiplication using short lattice vectors," in *Proc. 13th Algorithmic Number Theory Symp.*, 2019, pp. 293–310, doi: 10.2140/obs.2019.2.293.