Computer Networks 212 (2022) 109070

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A reliable and secure multi-path routing strategy for underwater acoustic sensor networks

Osman Gokhan Uyan *, Ayhan Akbas, Vehbi Cagri Gungor

O.G. Uyan, A.Akbas and V.C. Gungor are with Abdullah Gul University, Computer Engineering Dept., Kayseri, Turkey

ARTICLE INFO

Index Terms:
Data fragmentation
Encryption
Heuristics
Reliability
Security
Routing
Underwater acoustic sensor networks

ABSTRACT

Underwater Acoustic Sensor Networks (UASNs) have nowadays become an attractive topic in scientific studies and commercial applications. An important challenge in UASN's design is the limited network lifetime and low reliability caused by the limited battery energy of sensor nodes and harsh channel conditions in underwater environments. In addition, sensor nodes may generate sensitive data, which needs to be concealed. To this end, cryptographic encryption is a commonly used method to cipher a data before transmission to maintain security. However, encryption methods require additional computation and extra energy, which causes a decrease in the network lifetime. To this end, transmitting fragmented data through multiple paths can be used as a security countermeasure, in conjunction with encryption against silent listening attacks. To address these challenges, in this study, an optimization framework has been developed to analyze the effects of multi-path routing, packet duplication, encryption and data fragmentation on network lifetime. In addition to an optimal solution, Simulated Annealing, Golden Section Search and Genetic Algorithm-based heuristic methods have been developed. Performance results show that the proposed approach jointly solves the problem of UASN lifetime maximization, while providing network reliability and security.

1. Introduction

Surface of the earth is surrounded with water with a ratio of 3/4. Until recent years, the underwater world has been mostly undiscovered. Recent technologies developed in this area, such as Underwater Acoustic Sensor Networks (UASNs), allow new possibilities to explore and exploit underwater world for scientific studies and commercial applications. In a UASN, a number of nodes are distributed in a 3D space in order to track the underwater environment. The applications of a UASN can be for commercial or civil (i.e. exploring underwater life, monitoring pipelines) or military (i.e. invasion detection, mine detection) purposes. In UASNs, nodes communicate wirelessly using acoustic waves to transmit their sensing data to a sink node. This transmission may not always be directly, and intermediary nodes might act as relays to deliver the data coming from another node to the sink, in a multi-hop fashion.

The underwater nodes in the network operate using the power supplied by their limited batteries. In addition, limited network lifetime evoked by high transmission power requirements in harsh underwater environments is an important design issue for UASNs. Furthermore, most of the UASN applications demand high number of packet transmissions, which makes reliability and energy efficiency problem more arduous. Another important design issue, which needs to be handled is the security of the underwater network communications. Security is a critical issue for UASNs because nodes may generate sensitive data; they are deployed in an underwater space running autonomously and communication takes place in a broadcast medium. One type of attack is eavesdropping, where a silent listener tries to capture the transmitted data among the nodes without being caught [1]. To avoid the theft of sensitive data, the de facto application is encrypting the data before transmission, which renders the data meaningless unless it is decrypted. Encryption is a powerful technique for secure communication however it requires additional computing and thus, energy consumption, which is a disadvantage for limited-battery operated underwater nodes. In this study, our main motivation is to propose efficient solutions for the following important design issues:

- If the UASN application requires a certain network reliability, such as a certain packet success rate, how can we design a system that maintains this requirement? Moreover, what is the effect of

* Corresponding author.
E-mail address: gokhan.uyan@agu.edu.tr (O.G. Uyan).

https://doi.org/10.1016/j.comnet.2022.109070
Received 20 November 2020; Received in revised form 25 December 2021; Accepted 22 May 2022
Available online 26 May 2022

providing a reliability level on the energy consumption of the nodes under different network conditions?
- If the network application generates sensitive or secret data, what are the methods that are applicable to increase the security of the communication? Which encryption algorithms are more convenient to use in UASNs, and in more detail what is the impact of assigning a single encryption algorithm to all nodes in the network instead of choosing a logical algorithm for each node? What is the impact of encryption and data fragmentation on network lifetime and should these two methods be used jointly to increase system security?
- To address all abovementioned design issues, can we develop an optimization framework? Can we also use resource efficient meta-heuristic methods to approximately find optimal results? Which heuristic methods are more convenient to use for UASN optimization problems?

To address all these challenges and design issues, in this paper, an optimal multi-path routing strategy has been developed via mixed-integer programming (MIP) formulations to analyze the effects of multi-path routing, packet duplication, encryption and data fragmentation on network lifetime. The MIP model maximizes the lifetime of the most energy-starving sensor node while it guarantees a pre-determined reliability requirement. In addition, the AES and Twofish encryption algorithms have also been utilized to balance the trade-off between security supplied by encryption and network lifetime in UASNs. Although the proposed optimal multi-path routing strategy is modeled by using an MIP framework, the computational complexity arises from the nature of MIP encourages us to develop meta-heuristic solutions, since meta-heuristic approaches provide sub-optimal solutions in polynomial-time [3–6]. In order to overcome the computational complexity of the optimal multi-path routing strategy which is developed via MIP formulation, we develop different meta-heuristic approaches, namely, Golden Section Search (GSS), Simulated Annealing (SA), and Genetic Algorithm (GA). For each meta-heuristic algorithm, we investigate the near-optimal solution performance. To this end, the main contributions of our study can be summarized as follows:

- We developed a multi-path routing strategy via MIP formulations. The MIP model has the objective of maximizing the lifetime of the most energy hungry node while satisfying a pre-determined link reliability requirement. The MIP model captures the energy consumption trends of different encryption algorithms under the harsh channel conditions of underwater environments.
- There are some certain network applications like military surveillance where the network must maintain a requested data delivery rate in order to complete its duty successfully. In this study, we designed a network model which uses packet duplication in case of packet failures to provide a desired network success rate, i.e. desired network reliability. The nodes might send duplicated packets repeatedly over the same link or send them over separate paths.
- We proposed the usage of two symmetric encryption algorithms, AES and Twofish, throughout the network to secure the wireless communications. Against silent listening attacks, we proposed using data fragmentation and transmitting each fragment over separate links to make it harder for an adversary to collect all fragments and gather the entire data.
- We developed different meta-heuristic approaches, namely, Golden Section Search (GSS), Simulated Annealing (SA), and Genetic Algorithm (GA). For each meta-heuristic algorithm, we investigate the near-optimal solution performance.

The rest of this paper is structured as follows. Section II provides a background on related work in the literature. Section III presents detailed information about network model. Section IV gives brief information about encryption algorithms used. Section V presents performance results of the proposed optimization framework. Section VI gives implementation details and results of heuristic algorithms. Finally, Section VII concludes the paper.

2. Related work

UASNs generally operate in a multi-hop fashion, where contiguous nodes to the sink are used as relays during data transmissions. These nodes naturally spend more energy than further nodes since they make more transmission and reception operations. Routing procedures can help solving this problem by sharing the loads among the nodes more equally.

Cao et al. suggest a new transmission method named Energy Level Based Hybrid Transmission (ELT) that uses the remaining energy information of the nodes to decide using single-hop or multi-hop paths during transmissions [3]. By single-hop paths, the load of the nodes close to the sink are balanced. Multi-hop paths are used only if relay nodes have adequate remaining energy. ELT maintains better network lifetime when compared to single-hop paths only.

Su et al. present another routing method, which chooses the relay node based on a cost parameter for each link [4]. The cost parameter includes the remaining energy levels and the necessary transmission power between each connected node. They show that the algorithm increases network lifetime reasonably.

There is a trade-off between reliability and network lifetime since maintaining a reliability level requires more energy consumption. Some authors tried to find a balance between reliability and lifetime using routing algorithms. Pompili et al. present two routing methods to minimize energy consumption for delay-sensitive and delay-insensitive applications [5]. Both methods mitigate energy consumption by exploiting quality of the path to the next hop, necessary transmission power, and forward error correction rate. Using these parameters, the number of retransmissions needed for desired reliability level that cause higher energy consumption is evaluated.

Chen et al. blend reliability and lifetime in a routing algorithm. They propose an algorithm named Reliable and Energy Balanced Routing algorithm (REBAR), where a flexible packet transmission radius is set for the nodes according to their distance to the sink [6]. Nodes closer to the sink are given smaller radii to limit the possibility of them to become relays to avoid high energy consumption. While a small transmission radius is good for energy efficiency, it decreases the amount of successful packet delivery. By optimizing the parameters, the authors manage to provide energy efficiency and reasonable reliability during packet transmissions.

Using encryption for securing the transmitted data generated by UASNs is also studied by scientists. Xinbin et al. proposed an energy-efficient and secure data transmission method using encryption, based on chaotic compressive sensing (CCS) [7]. First, the method adopts compressive sensing (CS) using the sparsity of data in time domain. The method reduces transmission amount in a period by sampling the data in each frame and transmitting the data at the end of the period. Then, a CCS-based encryption is used to encrypt data at the end of a period to improve the security of transmission. They compare the proposed scheme with a conventional TDMA scheme and RACS scheme, and show that the proposed scheme improves bandwidth and energy consumption.

Castelluccia et al. propose a simple and secure additively homomorphic stream cipher to achieve efficient aggregation of encrypted data [8]. The proposed cipher uses modular additions and is suitable for CPU-constrained devices such as sensor nodes. They present that aggregation based on the proposed cipher can be used to efficiently compute statistical values such as mean, variance and standard deviation of sensed data, while achieving significant bandwidth gain and security.

Uluagac et al. introduce an energy-efficient Virtual Energy-Based Encryption and Keying (VEBEK) scheme for WSNs that reduces the number of transmissions for rekeying to avoid stale keys [9]. VEBEK

Table 1
Overview of the Related Work

Study	Channel Model	Multi-path	Min. reliability level	Encryption	Data fragmentation	Heuristics
Cao et al. [3]	✓	✓	×	×	×	×
Su et al. [4]	✓	×	×	×	×	×
Pompili et al. [5]	✓	×	✓	×	×	×
Chen et al [6]	✓	×	×	×	×	×
Xinbin et al [7]	✓	×	×	✓	×	×
Castelluccia et al. [8]	×	×	×	✓	×	×
Uluagac et al. [9]	×	×	×	✓	×	×
Incebacak et al. [1]	×	✓	×	×	✓	×
Lee et al. [11]	✓	✓	×	×	✓	×
Chen et al. [12]	×	✓	✓	×	✓	×
Xenakis et al. [13]	✓	×	×	×	×	✓
Yildiz et al. [14]	×	×	×	×	×	✓
Hosseini et al. [15]	×	×	✓	×	×	✓
Our study	✓	✓	✓	✓	✓	✓

encodes sensed data using a scheme based on a permutation code generated via the RC4 encryption. The key to the RC4 encryption changes as a function of the remaining virtual energy of the nodes. And, a one-time dynamic key is used for one packet only. They show that VEBEK is able to eliminate malicious data from the network in an energy-efficient manner with a 60-100% improvement in energy efficiency.

Multi-path routing with data fragmentation is another countermeasure that can be used against eavesdropping attacks. Moreover, multi-path routing provides several other favors, such as load balancing, reliability, and quality of service [10]. Incebacak et al. investigates the energy overhead of route diversity for security against node capture and eavesdropping attacks [1]. They developed an LP framework to model energy consumption and route diversity in WSNs. They conclude that energy overhead of route diversity increases with the level of security. If security is low, then the energy overhead is also low, and for high degrees of security, energy overhead can be huge.

Lee et al. studies the data distribution problem over multiple paths to minimize the maximum harm when a single link attack occurs [11]. The solution is formulated as a maximum-flow problem that can be solved in a distributed manner.

Chen et al. proposes a secure method for selecting multiple paths and transmitting data over these paths [12]. Their objective is to minimize the percentage of captured data by an adversary. Each path is assigned with a parameter that includes earlier performance on reliable data delivery. Multiple paths are selected based on these parameters and data is sent over these paths using min–max optimization and game theory.

Metaheuristic methods are invaluable for solving optimization problems in reasonable time with less resources than optimizations for UASNs. These methods strive for finding approximately optimal solutions. As mentioned in the previous section, WSN optimizations are often NP-hard problems which caused scientists to use metaheuristics in order to solve them. Xenakis et al. employ Simulated Annealing method to optimize network lifetime according to topology, transmission power and data packets [13].

Yildiz et al. propose an MIP optimization model to maximize network lifetime and maintain non-repudiation security [14]. They consider communication/computation energy characters of some Digital Signature (DS) algorithms. This problem is NP-hard, and they implement SA and GSS methods to solve it in reasonable time.

Hosseini et al. implement GA for the multi-objective optimization problem where reliability and power consumption are evaluated jointly [15]. Moreover, they proposed a Hierarchical Sub-Chromosome Genetic Algorithm (HSC-GA) to further decrease the solution time.

To the best of our knowledge, a study that jointly solves the problem of UASN lifetime maximization, while providing network reliability and security, has not been encountered in the literature yet. In this study, an optimization framework via MIP formulations has been developed to analyze the effects of multi-path routing, packet duplication, encryption and data fragmentation on network lifetime, while providing network reliability. To overcome the computational complexity of the optimal multi-path routing strategy, we also develop different meta-heuristic approaches, namely, Golden Section Search (GSS), Simulated Annealing (SA), and Genetic Algorithm (GA). For each meta-heuristic algorithm, we also investigate the near-optimal solution performance. An overview of the given related work is presented in Table 1.

3. Network model

The network model used in this paper is composed of randomly distributed nodes in 3D underwater space. In the UASN, all nodes except the sink generate data packets as they sense the environment. These packets are transmitted to the sink node over multi-hop paths. Since the sensor nodes operate in an autonomous fashion, we assume there is no data delivery from the sink to the sensor nodes. Moreover, we employ a TDMA based approach for communication, where each node sends its data inside the time slot allocated for it, so we assume there is no interference among the communications. Here, it is important to note that network security can be augmented by using multi-path routing along with encryption. Transmitting fragmented data, instead of entire data, through multiple paths to the sink node can be used as a security countermeasure in conjunction with encryption. Although encryption is a good defense against silent listening attacks, multi-path routing renders eavesdropping more difficult and helps improve the network security [2]. By slicing the data into fragments and transmitting each fragment through different paths, we force an adversary to capture all fragments to be able to reconstruct a node's data; also since we are sending duplicated packets in order to increase the reliability of the network, it can be easier for an adversary to listen any of the links along which the duplicated data is transmitted. Hence, data fragmentation comes out as an efficient solution for the security weakness caused by packet duplication. Moreover, gathering fragments of data from separated links may demand more energy consumption compared to gathering data from a single path. Particularly, the adversary has to consume more resources in order to get all the data fragments to reconstruct the complete data if multi-path routing is employed. An important point to stress here is that, multi-path routing not only makes the adversaries spend more energy but also it brings an energy burden to the UASN. If the data is fragmented and transmitted through multiple paths, it is certain that all data cannot be transmitted using optimal paths. Some piece(s) of the data needs to be transmitted towards less efficient paths, which will in return, cause some decrease in the network lifetime.

In the proposed optimization framework, the network needs to satisfy a reliability threshold named network success rate

(NSR), which defines the minimum necessary expectation of a generated packet to reach the sink node successfully, that is packet delivery ratio. In the harsh environment of underwater, where the possibility of successful transmission on a single path is lower than the

$$\text{minimize } \theta \tag{5}$$

subject to:

$$\theta \geq t_p \sum_{k=1}^{n} P_k \left(\sum_{\ell \in \delta^+(j)} P_{r,\ell} y_{\ell k} + \sum_{\ell \in \delta^-(j)} P_{t,\ell} x_{\ell k} \right) + \sum_{\ell \in \delta^+(j)} E_k x_{\ell k} \ , \forall j \in N \backslash \{0\}, \tag{6}$$

$$M \sum_{\ell \in \delta^+(j)} y_{\ell k} \geq \sum_{\ell \in \delta^-(j)} x_{\ell k} \ , \forall j \in N \backslash \{0\}, \forall k \in N \backslash \{0\}, j \neq k, \tag{7}$$

$$M \, u_{ik} \geq x_{ik} \ , \forall i \in A, \forall k \in N \backslash \{0\}, \tag{8}$$

$$\sum_{\ell \in \delta^-(j)} x_{\ell k} \leq M r_a, \forall j \in A, \forall k \in N \backslash \{0\}, \tag{9}$$

$$\sum_{\ell \in \delta^+(j)} y_{\ell k} \leq \sum_{\ell \in \delta^-(j)} x_{\ell k} \ , \forall j \in N \backslash \{0\}, \forall k \in N \backslash \{0\}, j \neq k, \tag{10}$$

$$\sum_{\ell \in \delta^-(j)} x_{\ell k} \geq 1, \forall k \in N \backslash \{0\}, \tag{11}$$

$$x_{ik} = M y_{ik}, \forall i \in A, \forall k \in N \backslash \{0\}, \tag{12}$$

$$\sum_{\ell \in \delta^+(j)} y_{\ell k} \geq r_a, \forall k \in N \backslash \{0\}, \tag{13}$$

$$\sum_{k=1}^{n} P_k \left(\sum_{\ell \in \delta^+(j)} y_{\ell k} + \sum_{\ell \in \delta^-(j)} x_{\ell k} \right) \leq t_r, \forall j \in N \backslash \{0\}, \tag{14}$$

$$v_{t_i k} - v_{h_i k} + n u_{ik} \leq n - 1, \forall i \in A, \forall k \in N \backslash \{0\}, \tag{15}$$

$$x_{ik} \geq 0, \forall i \in A, \forall k \in N \backslash \{0\}, \tag{16}$$

$$y_{ik} \geq 0, \forall i \in A, \forall k \in N \backslash \{0\}, \tag{17}$$

$$u_{ik} \in \{0,1\}, \forall i \in A, \forall k \in N \backslash \{0\}, \tag{18}$$

$$\theta \geq 0, \forall i \in A, \forall k \in N \backslash \{0\}, \tag{19}$$

$$\sum_{\ell \in \delta^+(j)} y_{\ell k} \leq L_{node}, \forall k \in N \backslash \{0\}. \tag{20}$$

Fig. 1. Optimization model

threshold, packets need to be duplicated, and sent through the same path multiple times, or sent through multiple paths synchronously. The decision for the amount of duplication and the paths to be used is based on packet success rate (PSR), which is calculated by for a given packet length of l and bit error rate (BER):

$$PSR = (1 - BER)^l \tag{1}$$

If a node transmits a packet to another node over a single path, PSR for the utilized path is equal to the network success rate (NSR) for that packet. If packet duplication is used, NSR for the transmitted packet from a node is calculated based on PSR of all the links where an individual packet is transmitted on:

$$NSR = 1 - \prod_{i \in D} (1 - PSR_i) \tag{2}$$

where D is the collection of duplicate packets and PSR_i is the PSR of the i^{th} path where the packet is sent over. Bit Error Rate (BER) is fixed for all nodes, which makes PSR values same for all paths. Thus, whether all the packets are sent through the same path or through multiple paths does not affect NSR calculation. In the case of sending duplicate packets over n multiple paths or n duplicate packets over one path, the NSR becomes:

$$NSR = 1 - (1 - PSR)^n \tag{3}$$

Using (3), the minimum number of duplication rate of a packet to satisfy a selected NSR threshold is calculated by:

$$M = \frac{\log(1 - NSR)}{\log(1 - PSR)} \tag{4}$$

At this point, note that each relay node might duplicate the data coming from the source node according to the NSR between itself and the sink. The decision of employing multiple paths or not is important for load balancing among the nodes and is taken by the optimization model. The proposed model minimizes the maximum energy spent by the nodes in a single round. The problem of maximizing the network lifetime is solved by minimizing the maximum expected energy consumed by the nodes in a single transmission round. A transmission round can be described as the time period needed for all of the packets generated by all of the nodes to arrive at the sink node. The optimization model is given in Fig. 1.

In the model, objective function (5) is given to minimize the maximum energy consumption. (6) states that θ has to be greater than or equal to the energy consumption of each node calculated by transmission, reception and encryption energies. (7) provides that if a relay does not receive a packet from node k, then it cannot transmit anything originating from node k; (8) and (18) ensure that if there is traffic on a path, then it is occupied; (9) limits packet duplication that will restitute

only the losses of the next path. (10) supports packet duplication and confirms that total outbound traffic from a relay is at least as great as the total inbound traffic; and (11) confirms that at least one packet should go out from original nodes, which approves packet duplication if it leads to a better network lifetime. (12) models packet failures, while (13) ensures that NSR is achieved. (14) limits the number of packets a node can send and receive to the period of a single round (t_r). (15) confirms that

$$minimize\ \theta \tag{5}$$

subject to :

$$\theta \geq\ t_p \sum_{k=1}^{n} P_k \left(\sum_{\ell \in \delta^+(j)} P_{r,\ell} y_{\ell k} + \sum_{\ell \in \delta^-(j)} P_{t,\ell} x_{\ell k} \right) + \sum_{\ell \in \delta^+(j)} E_k x_{\ell k},\ \forall j \in N \backslash \{0\}, \tag{6}$$

$$M \sum_{\ell \in \delta^+(j)} x_{\ell k} \geq \sum_{\ell \in \delta^-(j)} x_{\ell k},\ \forall j \in N \backslash \{0\},\ \forall k \in N \backslash \{0\},\ j \neq k, \tag{7}$$

$$M\ u_{ik} \geq x_{ik},\ \forall i \in A,\ \forall k \in N \backslash\ \{0\}, \tag{8}$$

$$\sum_{\ell \in \delta^-(j)} x_{\ell k} \leq M r_a,\ \forall j \in A,\ \forall k \in N \backslash\ \{0\}, \tag{9}$$

$$\sum_{\ell \in \delta^+(j)} y_{\ell k} \leq \sum_{\ell \in \delta^-(j)} x_{\ell k},\ \forall j \in N \backslash \{0\},\ \forall k \in N \backslash \{0\},\ j \neq k, \tag{10}$$

$$\sum_{\ell \in \delta^-(j)} x_{\ell k} \geq 1,\ \forall k \in N \backslash \{0\}, \tag{11}$$

$$y_{ik} = M y_{ik}, \forall i \in A,\ \forall k \in N \backslash \{0\}, \tag{12}$$

$$\sum_{\ell \in \delta^+(j)} y_{\ell k} \geq r_a,\ \forall k \in N \backslash \{0\}, \tag{13}$$

$$\sum_{k=1}^{n} P_k \left(\sum_{\ell \in \delta^+(j)} y_{\ell k} + \sum_{\ell \in \delta^-(j)} y_{\ell k} \right) \leq t_r, \forall j \in N \backslash \{0\}, \tag{14}$$

$$v_{t_i k} - v_{h_i k} + n u_{ik} \leq n - 1, \forall i \in A, \forall k \in N \backslash \{0\}, \tag{15}$$

$$x_{ik} \geq 0, \forall i \in A, \forall k \in N \backslash \{0\}, \tag{16}$$

$$y_{ik} \geq 0,\ \forall i \in A, \forall k \in N \backslash \{0\}, \tag{17}$$

$$u_{ik} \in \{0,1\}, \forall i \in A, \forall k \in N \backslash \{0\}, \tag{18}$$

$$\theta \geq 0,\ \forall i \in A, \forall k \in N \backslash \{0\}, \tag{19}$$

$$\sum_{\ell \in \delta^+(j)} y_{\ell k} \leq L_{node},\ \forall k \in N \backslash \{0\}. \tag{20}$$

there are no cycles in the routes; it is acclaimed as MTZ constraints [16]. (16), (17), and (19) ensure the nonnegativity of the decision variables and (18) ensures that u_{ik} variables are binary. Finally, (20) is given for packet fragmentation, the sum of all flows carrying node k's data to node i is limited by L_{node}, so that only a fragment of the data of node k is assured to reach the relay. For instance, if $L_{node} = 0.5$, we limit the flows reaching a relay such that it can receive at most 50% of the source node's packets, which portends that source node has to divide its packet into two fragments. Similarly, $L_{node} = 1$ implies that there is no limitation for a relay and it can receive a whole packet submitted by the source node.

The explanation of the symbols used in Fig. 1 is as follows:

Sets:

N : Set of the sensor nodes.
A : Set of the links between the nodes (arcs).
$\delta^+(j)$: Set of arcs that are incoming to node j.
$\delta^-(j)$: Set of arcs that are outcoming from node j.

Indices:

i : Indexes of set A; $i \in 1, \ldots, m$.
j : Indexes of set N; $j \in 0, \ldots, n$.
k : Index of the source node; $k \in N \backslash \{0\}$ implies that the sink cannot be a source (no data outcomes from the sink).
ℓ**:** Indexes sets $\delta^+(j)$ and $\delta^-(j)$.

Parameters:

$P_{t,\ell}$: Transmission power spent over arc ℓ.
$P_{r,\ell}$: Reception power spent over arc ℓ.
E_k : Encryption energy spent by node k.
t_r : Duration of 1 transmission round.
t_p : Duration needed for transmission of 1 packet.
h_i : Receiving node of arc i (head).
t_i : Transmitting node of arc i (tail).

Decision Variables:

x_{ik} : Fraction of packets coming from node k to node i. $i = (u, v)$ implies that u transmits $100^* x_{ik}$packets to v.
y_{ik} : Fraction of packets coming from node k received by node i. $i = (u, v)$ implies that v received $100^* y_{ik}$packets from u (and $x_{ik} - y_{ik}$ is lost because of bit errors).
u_{ik} : Equals to 1 if arc i is occupied and 0 otherwise.
v_{jk} : Used to break any cycles. It implies position of node j on the path between node k and the sink.
θ : Maximum amount of expected energy consumption of nodes 1 to n.

For the underwater channel model, the essential issue about evaluating lifetime of the network is calculating the transmission power of the sensor nodes. Since the underwater environment has harsh channel conditions, transmission power depends highly on transmission distance. Felemban et al. uses necessary intensity (I_t) in 1 meter distance to find the required transmission power [26] as follows:

$$P_t = I_t \times 2\pi \times 1m \times h \tag{21}$$

Reference intensity is calculated by:

$$I_t = 10^{\frac{SL(d,f)}{10}} \times I_0 \tag{22}$$

Evaluating Source Level (SL) requires Signal to Noise Ratio (SNR), attenuation (A) and noise (N) are known.

$$SL(d,f) = A(d,f) + N(f) + SNR \tag{23}$$

To evaluate A, N and SNR, these formulas are used:

$$A(d,f) = \kappa \log(d) + \alpha(f) \times d \times 10^{-3} \tag{24}$$

$$N(f) = N_t(f) + N_s(f) + N_{th}(f) + N_w(f) \tag{25}$$

$$SNR = 10^{\frac{SNR(d,f)}{10}} \tag{26}$$

SNR (in non-dB) is evaluated according to SNR per bit value (E_b/N_0), data rate and noise bandwidth.

$$SNR(d,f) = \frac{E_b}{N_0} \times \frac{R}{B_N} \tag{27}$$

and E_b/N_0 is found using:

$$BER = \frac{3}{8} erfc\left(\sqrt{\frac{4}{10} \times \frac{E_b}{N_0}}\right) \quad (28)$$

For each link, reception power for the target node is assumed as 1/10 of the transmission power that the source node uses.

4. Encryption algorithms

There are two basic types of encryption schemes named as Symmetric Key Encryption and Public Key Encryption. As its name implies, in symmetric key encryption schemes, the key that is used for encryption and decryption are the same. The sides that are communicating need to obtain the same key in order to successfully complete the encryption and decryption operations. AES, Blowfish, Twofish, DES, 3DES, RC5 and RC6 are some of the example cryptography algorithms that use symmetric key schemes. In public key encryption schemes, there are two different keys for encryption and decryption operations. The key used for encryption is published for anyone who is willing to encrypt plaintext with that key. On the other hand, the key that is necessary to decrypt the ciphertext is only in the hands of the intended receiver to be able to decrypt and read the plaintext. The most conversant cryptography algorithms that use public key schemes are RSA and Elliptic Curve Cryptography.

Encryption has been used for both military and civil purposes for a long time to maintain secure communication. In our study, we use encryption to secure the data that is in transit, against the type of attacks that is called silent listening or eavesdropping network traffic by unauthorized parties. When we have assessed which type of encryption scheme to use, we ascertained that it is more convenient to use a symmetric key scheme in a UASN. Because the nodes in the network are determined before deploying them underwater, and we do not need to publish a public encryption key throughout the network after deployment. Moreover, key distribution is an important issue in WSN design and it is another topic independent of our study. Instead of distributing keys, we place the encryption key -which is different for each node- into the nodes' memory before deployment and we use that key for encryption operations during the lifetime of the network. Only the sensed data is encrypted and the communication stack headers, including routing data, are clear so that routing operations can be completed. The sink owns the whole set of the encryption keys of each sensor, and relay nodes cannot decrypt the message coming from other nodes, which increases security under node capture too. A point to be stressed here is that, under a node capture attack the symmetric key can be captured too, however, if the node can be captured, the sensor data that it produces can be read directly without the need of decryption. Another reason we select symmetric key encryption scheme is that, despite providing privacy and easy key management, public key algorithms make encryption slower and need very intensive computation [17]. RSA algorithm has been evaluated that it is 10 to 10000 times slower than DES algorithm in various environments and it needs more computational energy which is not a desired condition in our situation. Based on these considerations, the selected encryption algorithms have been explained in the following subsections.

4.1. AES

AES is a specific subset of the Rijndael block cipher, with a block size of 128-bits and key sizes of 128, 192 and 256-bits [18]. In 2002, it was chosen to become the US federal standard and it became a de facto encryption algorithm since then. The reason we selected AES is that, it is stronger and faster than its predecessor 3DES, it is relatively simple to implement and it can be implemented in both software and hardware easily. Since the encryption operation will be done on the hardware of the nodes, it is convenient to select this algorithm. At this point, we assume that the implementation of AES is preloaded on the nodes before deployment, and it can be executed programmatically before transmission of a data. We select 128bit keys for AES, which is more than enough for maintaining security of the encrypted data. A 128bit-key means $3.4*10^{38}$ possible combinations which is nearly impossible to be cracked using brute-force attacks. Moreover, using a larger sized key makes the implementation more complex, increases the duration of the operation and energy consumption as a result.

AES relies on the design principle known as substitution-permutation network, which comprises of linked operations that involve replacing inputs by specific outputs (substitutions) and shuffling bits around (permutations). It uses a fixed, 128bit block size and one of the key sizes of 128, 192 or 256 bits [18]. The selected key size indicates the number of transformation rounds, 10, 12 and 14 rounds for 128, 192 and 256bit keys respectively. Every round consists of some processing steps, with one step that depends on the encryption key itself. A set of reverse rounds are applied to decrypt the ciphertext using the same encryption key. Note that, nodes on the network will not make any decryption, though decryption will be done on the sink node or base station which do not have any constraint about energy consumption.

4.2. Twofish

In this study, we include Twofish [19] as a second encryption algorithm. Twofish also uses one of the key sizes of 128, 196 or 256bits, and a block size of 128bits. The Twofish algorithm was first introduced in 1998, and has not been cracked yet [20]. On the other hand, it is more lightweight than AES128 and it consumes less energy. The algorithm uses pre-computed key-dependent S-boxes, and a relatively complex key schedule. One half of an key is used as the encryption key while other half of the key is used to modify the encryption algorithm (key--dependent S-boxes). It is slightly slower than AES when 128bit keys are used while it is faster when 256bit keys are used. This algorithm is not patented, it is placed in the public domain.

4.3. Energy consumption calculation

In the network architecture, we use the micromodems developed by Gangneung-Wonju National University, Korea as sensor nodes and we let every sensor node is attached with a camera that generates 87*143*3 = 37323 bytes of raw photographic data [21]. In order to measure the speed performance of the AES and Twofish, we use the proposed technique given in [22]. The algorithms are coded in C language in accordance with their definitive documents for their speed performances. To obtain the CPU runtime values for the algorithms, all unused CPU peripherals, interrupt routines, timers, and ports are disabled before algorithms are run. The encryption process of each algorithm is executed 1000 times in an infinite loop to avoid the looping latency in the results. Moreover, in order not to introduce a delay, a logic analyzer is employed to measure the runtime of the algorithms electronically through the CPU port pin which is toggled in the encryption loop, yielding the square waveform generated on the pin. The measured wave period is divided by 1000 to obtain the averaged pure CPU time for the encryption process. After the experiments, execution runtimes of the algorithms are measured as 64.601 msec for AES and 31.549 msec for Twofish algorithms. Selected micromodems draw 17.4mA during the encryption process, consuming 57.42mW of power. As a result, energy requirements for the complete AES and Twofish encryption operations are calculated as 3.710 mJ and 2.038 mJ respectively.

Note that the micromodems are used in an "as is" fashion, we do not consider any modification like adding a second board to the configuration for heavy computation duties because this needs both changing the software of the modem as well as it brings another hardware cost. Moreover, the new hardware will still need a separate battery to operate, and energy consumption of a new configuration needs a deeper study which is out of the scope of this paper.

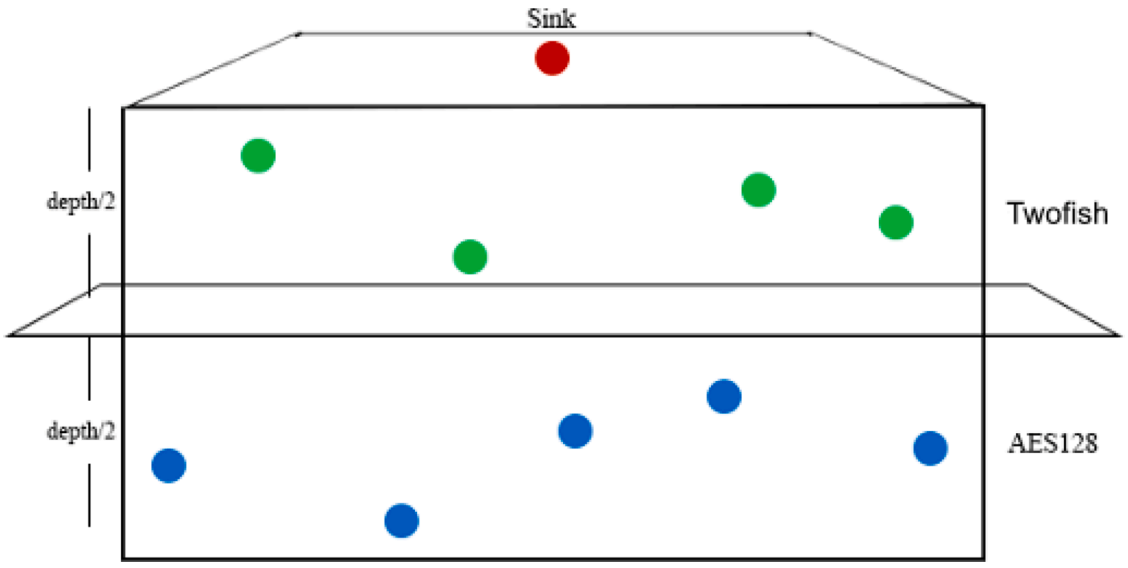

Fig. 2. Encryption decision strategy

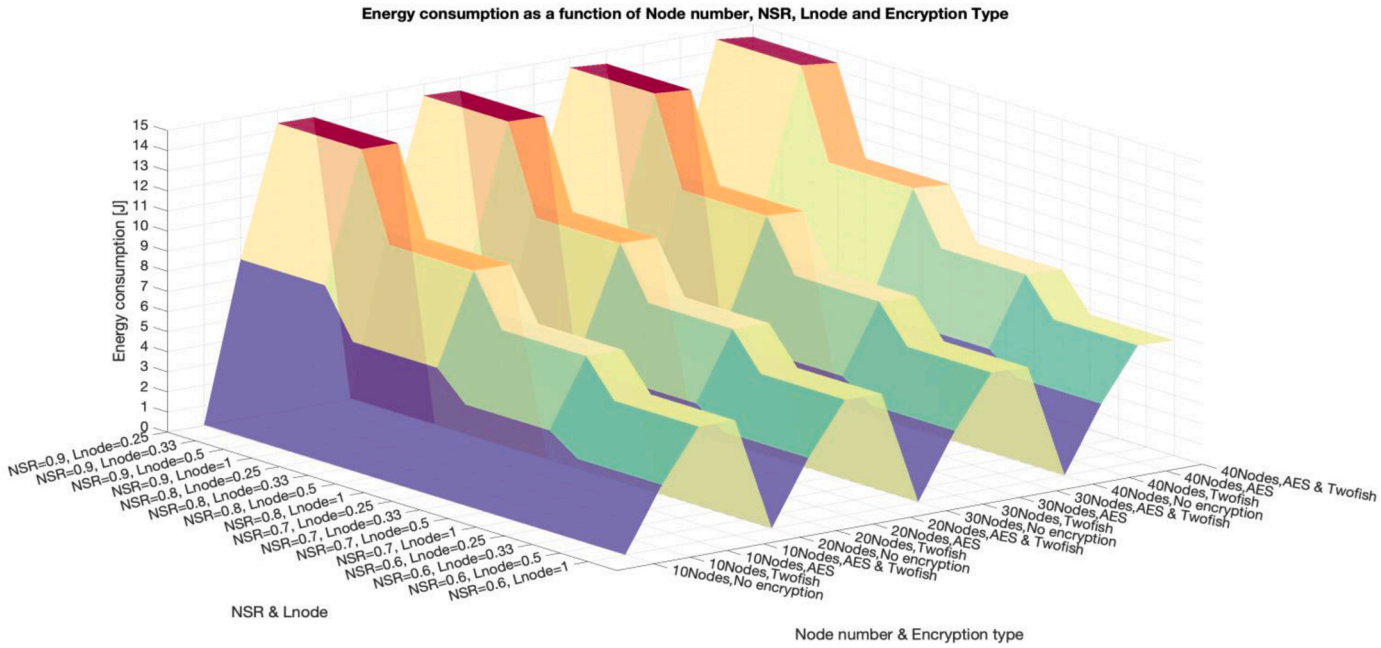

Fig. 3. Energy consumption as a function of node number, NSR, L_node and encryption type

4.4. Encryption decision

For the decision of which encryption algorithm to be used, we consider the depth of each node in 3D space. If the depth of a node is smaller than half of the total depth of the network, it uses Twofish, otherwise it uses AES128 for encryption. Fig. 2 shows a sample configuration for a 10-node network.

The reason behind this decision is that, energy consumption of Twofish is smaller than AES, and the nodes that are closer to sink consume more energy than further ones since they mostly relay other nodes' packets along with their own data. By using Twofish with less distant nodes, we still aim to provide security for the data they produce, and we allow them to spend lower energy for encryption operation. Please note that for encryption, we have selected 4 options; no encryption, network-wide encryption with Twofish, network-wide encryption with AES, and AES-Twofish mixed encryption according to the idea given in Fig. 2.

5. Performance results

The results presented in this work are gathered using MATLAB software [23] for generating network parameters, and CPLEX solver [24] for solving the optimization model. Optimization solver calculates net energy consumption values for each single data generation period. The maximum consumption among all the nodes is represented as θ. During each period, every node but sink generates a single packet. To handle high number of relay packets caused by packet duplication, period duration is set long enough to allow transmissions of up to 500 packets. Size of each packet is determined as 10.000 bytes.

The main aim of this study is to analyze effects of reliability as well as security on the network lifetime. The network parameters are chosen accordingly: The simulations are run with 10, 20, 30 and 40 nodes, each

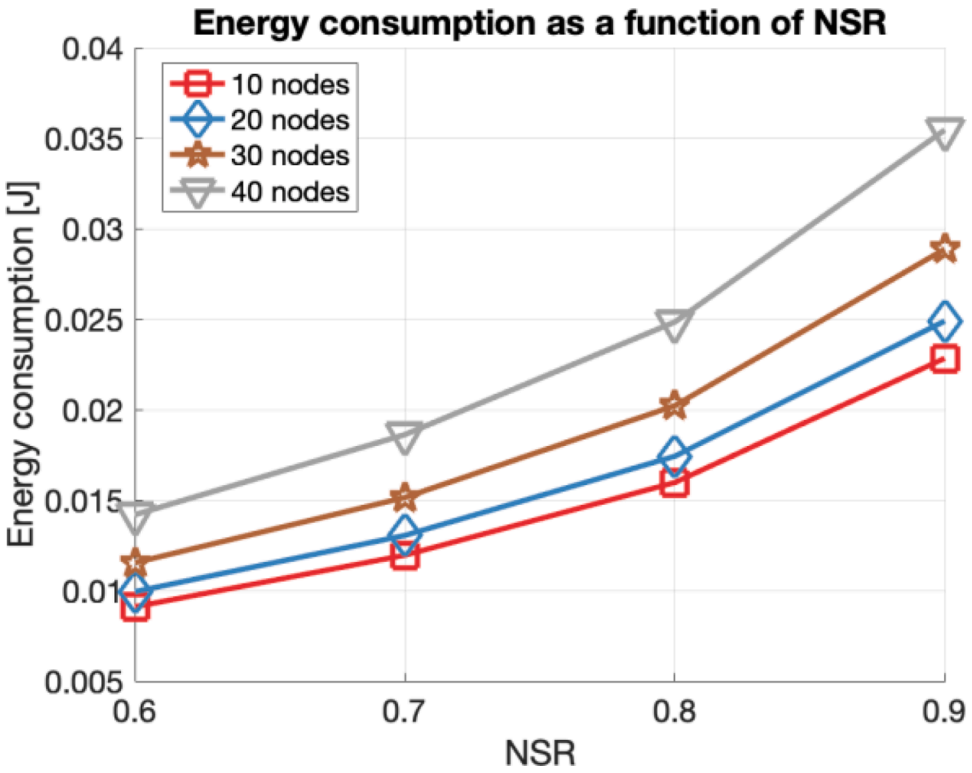

Fig. 4. Energy consumption as a function of NSR

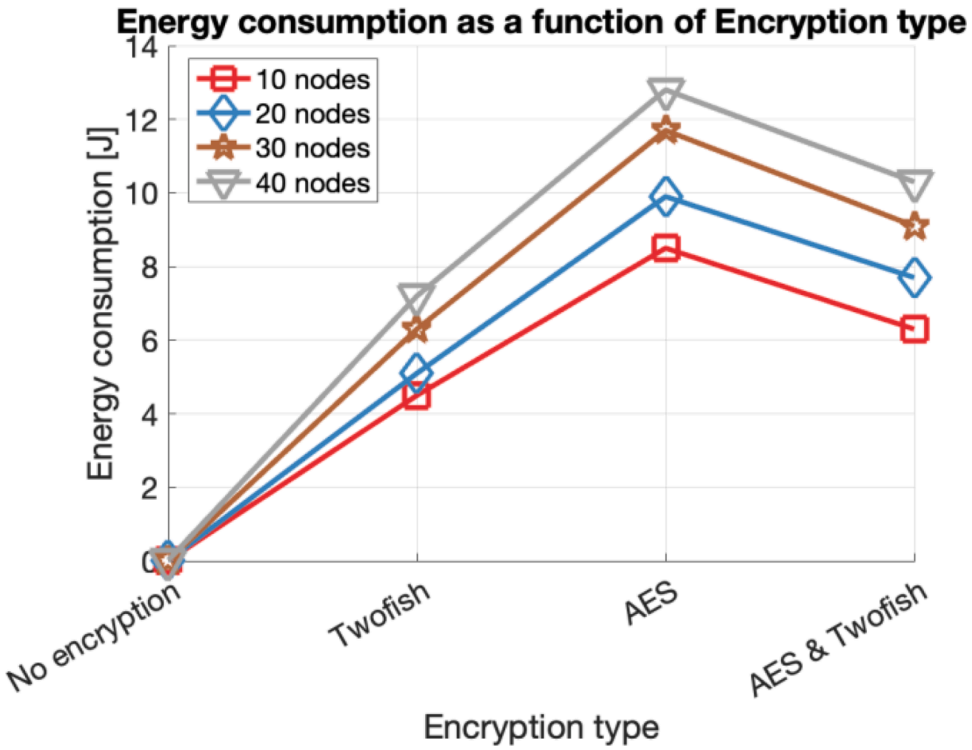

Fig. 5. Energy consumption as a function of encryption type

with 100 randomly chosen topologies. For simplicity, we have chosen relatively small network sizes which are sufficient to demonstrate the functionality of the proposed network model. Running simulations with larger sized networks takes too much time to complete and in some cases the optimizer cannot come up with a feasible solution within an acceptable duration. Moreover, no assumption is made about the distribution of the nodes. Instead, we are investigating to arrive at a general understanding of the performance of the suggested algorithms by choosing different random topologies for each network size. NSR is taken as 0.6, 0.7, 0.8 and 0.9 in the simulations. NSR is not a computed

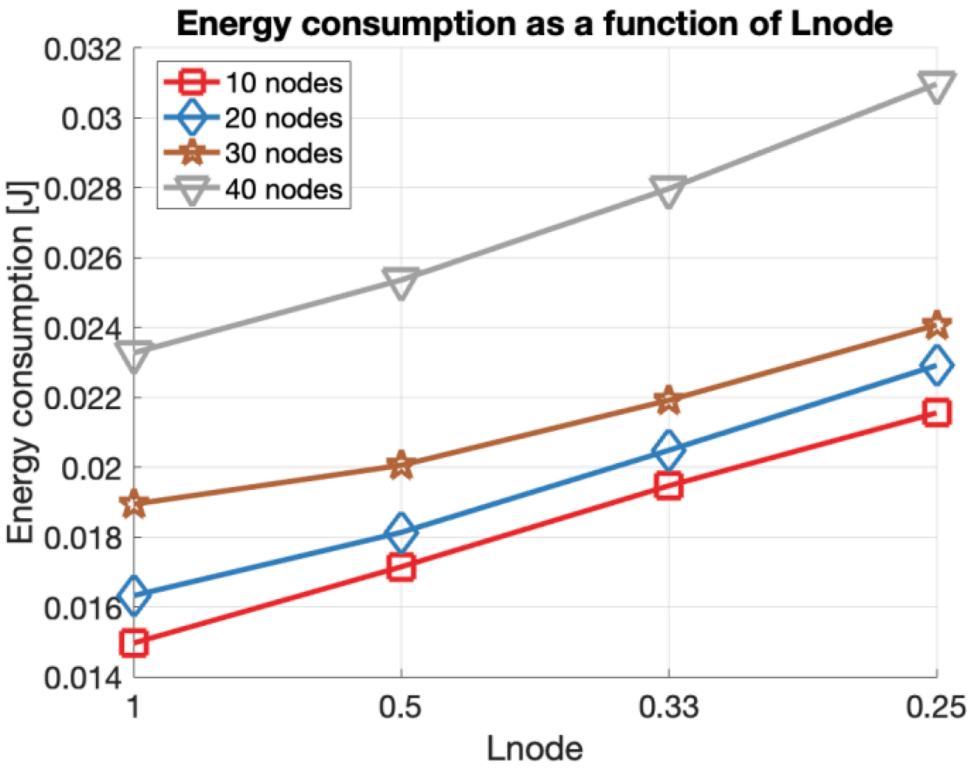

Fig. 6. Energy consumption as a function of L_node

Table 2
Simulated Annealing Algorithm

Pseudocode for Simulated Annealing Algorithm	
Input:	Number of the nodes, initial temperature t, cooling parameter α, and maximum iteration number *maxite*.
Output:	Minimum energy consumption of the highest energy consuming node, best solution.
1:	Initialize parameters; Iteration ($ite \leftarrow 0$), Best Energy ($bestE \leftarrow 0$);
2:	Find initial solution and set it to current and best solutions (*curSol* and *bestSol*). Find the energy consumption using current solution, set the result as current energy and best energy (*curE* and *bestE*);
3:	**while** $ite <= maxite$ **do**
4:	Assign temperature ($t \leftarrow t \times \alpha$). Find a candidate solution (*candSol*) by reversing one random element in *curSol*. Find its solution as candidate energy (*candE*);
5:	**if** $candE < curE$ **then**
6:	$curE \leftarrow candE$;
7:	$curSol \leftarrow candSol$;
8:	**if** $candE < bestE$ **then**
9:	$bestE \leftarrow candE$;
10:	$bestSol \leftarrow candSol$;
11:	**end if**
12:	**else**
13:	$\Delta Obj \leftarrow curE - bestE$. Assign $candSol \leftarrow curSol$ with probability: $e^{-(\Delta Obj/t)}$.
14:	**end if**
15:	$ite \leftarrow ite + 1$;
16:	**end while**
Return:	*bestE, bestSol.*

Table 3
Golden Section Search Algorithm

Pseudocode for Golden Section Search Algorithm	
Input:	Number of the nodes (n), lower bound for interval ($lb \leftarrow 0$), upper bound for interval ($ub \leftarrow n$ -1), and golden ratio ($\Phi \leftarrow (-1+\sqrt{(5)})/2$).
Output:	Minimum energy consumption of the highest energy consuming node, best solution.
1:	Compute $\lambda_1 \leftarrow \lceil (ub - \Phi \times (ub - lb)) \rceil$ and $\lambda_2 \leftarrow \lfloor (lb + \Phi \times (ub - lb)) \rfloor$;
2:	**while** $\lvert ub - lb \rvert >= 1$ **do**
3:	Assign AES to first $\lambda_1 + 1$ nodes, and Twofish to remaining nodes for encryption. Estimate the energy consumption as E_1;
4:	Assign AES to first $\lambda_2 + 1$ nodes, and Twofish to remaining nodes for encryption. Estimate the energy consumption as E_2;
5:	**if** $E_2 < E_1$ **then**
6:	$ub \leftarrow \lambda_2$; $\lambda_2 \leftarrow \lambda_1$; $\lambda_1 \leftarrow \lceil ub - \Phi \times (ub - lb) \rceil$; $bestE \leftarrow E_2$;
7:	**Else**
8:	$lb \leftarrow \lambda_1$; $\lambda_1 \leftarrow \lambda_2$; $\lambda_2 \leftarrow \lfloor lb + \Phi \times (ub - lb) \rfloor$; $bestE \leftarrow E_1$;
9:	**end if**
10:	**end while**
Return:	$bestE, AES \leftarrow 2 <= i <= lb$, *Twofish* $\leftarrow lb+1 <= i <= n$.

parameter and we select these values by assuming network scenarios that need these network success rates. For data fragmentation, L_{node} is selected as 1.0 (no limitation), 0.5, 0.33, and 0.25, which means a node splits its data at least into 2, 3 and 4 fragments before transmission respectively. Note that, in the simulations, we neglect the packet overhead caused by data fragmentation, since they appear to be too small (a few bytes) compared to packet size. For encryption, we again select 4 options; no encryption, network-wide encryption with Twofish, network-wide encryption with AES, and AES-Twofish mixed encryption according to the idea given in Fig. 2. The final energy consumption results are found by taking average of 100 different results found for 100 randomly generated network topologies for each node number.

Fig. 3 represents the energy consumption as a function of node number, encryption type, NSR and L_node. Fig. 3 presents that, an increase in the node number also increases the energy consumption of the nodes. The main reason behind this is, more nodes generate and transmit more data and the relays in-between need to make more reception and transmission operations in large-sized networks. Since transmission and reception operations consume high energy in UASNs, larger networks tend to spend more energy. As for encryption types, using only Twofish algorithm causes a small decrease in network lifetime, while using only AES causes a higher decrease in network lifetime. When AES and Twofish algorithms are used together, encryption energy is demarcated and the system still reaches the desired security level. For different NSR values, we can understand that forcing the system to obtain a higher reliability threshold causes a decrease in the network lifetime.

Table 4
Genetic Algorithm

Pseudocode for Genetic Algorithm	
Input:	Size of population (*pSize*), maximum number of generations (*maxGen*), probability of mutation (*pMut*).
Output:	Minimum energy consumption of the highest energy consuming node, best solution.
1:	Initiate parameters; Current generation ($curGen \leftarrow 0$), generate population ($P \leftarrow \{C_1, \ldots, C_{pSize}\}$), evaluate best energy for population P ($bestE \leftarrow bestE(P)$).
2:	**while** $curGen <= maxGen$ **do**
3:	Pick two chromosomes using Roulette Wheel function; $\{C_{c1}, C_{c2}\} \leftarrow rouletteWheel(P)$;
4:	Crossover the picked chromosomes; $C_{new} \leftarrow crossOver(C_{c1}, C_{c2})$;
5:	Mutate C_{new} with probability *pMut*; $C_{new} \leftarrow mutate(C_{new}, pMut)$;
6:	Evaluate the energy consumption for C_{new} as *curE*.
7:	**if** $curE < bestE$ **then**
6:	Assign new chromosome as best solution; $bestSol \leftarrow C_{new;}$
7:	Pick the chromosome giving highest energy consumption; $C_h \leftarrow pickChromosome(P)$;
8:	Extract C_h from *P*; $P.extract(C_h)$;
9:	Include new chromosome to *P*; $P.include(C_{new})$;
10:	**end if**
11:	$curGen \leftarrow curGen + 1$;
12:	**end while**
Return:	*bestE, bestSol.*

Increasing NSR causes more packet duplications and more data transmissions, some of which are on less optimal paths, and energy consumption increases accordingly. Increasing the number of data fragments causes more transmission and reception operations which causes the increase in the energy consumption. On the other hand, when data fragmentation is used along with encryption, it causes only 1-2% increase on the energy consumption which makes it logical to be used along with encryption to improve the security of the network.

Fig. 4 represents performance results as a function of NSR only, for 10-40 nodes. In this scenario no encryption is used and L_node=1. As it can be seen from the figure that, increasing NSR increases energy consumption visually. Moving NSR from 0.6 to 0.9 causes around 200% increase in terms of energy consumption. This is an expected result because by increasing the NSR, we force the system to make packet duplications and as a result more transmission and reception operations some of which are on sub-optimal paths.

Fig. 5 represents simulation results as a function of encryption type only, for 10-40 nodes. In this scenario NSR is selected as 0.6 and L_node=1. As the figure represents, using Twofish throughout the network causes about 4-7 Joules increase in terms of energy consumption while this amount is about 8-13 Joules for AES algorithm. Using AES and Twofish together delimits the increase in energy consumption while supplying a desired security level.

Fig. 6 represents simulation results as a function of L_node only, for 10-40 nodes. Again, in this scenario NSR is selected as 0.6 and no encryption is applied. As the figure represents, increasing the number of data fragments increases the energy consumption of the network. For different L_node values, we can understand that forcing the system to make more fragmentation causes a decrease in the network lifetime. Because, increasing fragmentation causes more transmissions, some of which are on less optimal paths, and energy consumption increases accordingly.

6. Heuristic approaches

After gathering results of optimizations, we decided to implement some well-known heuristic algorithms to ease the solution of the network model, because solving optimization models can take seriously high amount of computing and resources, and even in some cases optimizations cannot come up with a feasible solution within an acceptable

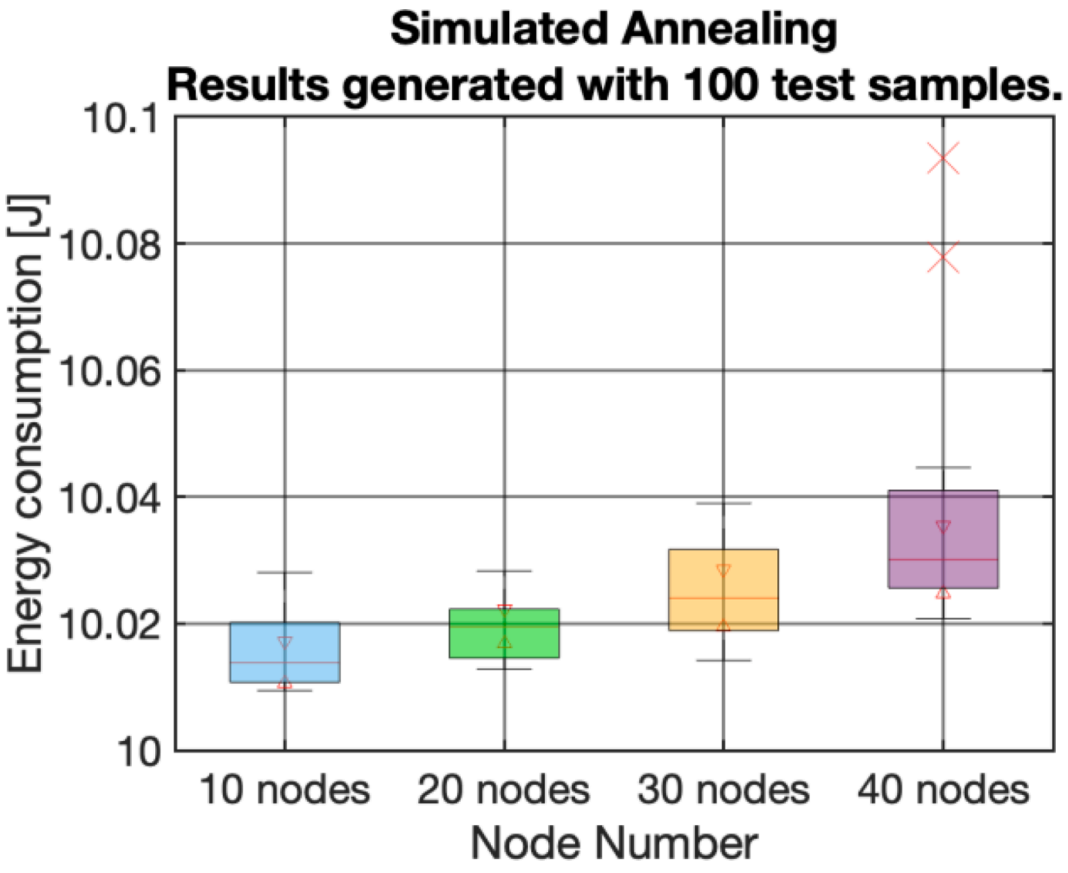

Fig. 7. Results of Simulated Annealing algorithm.

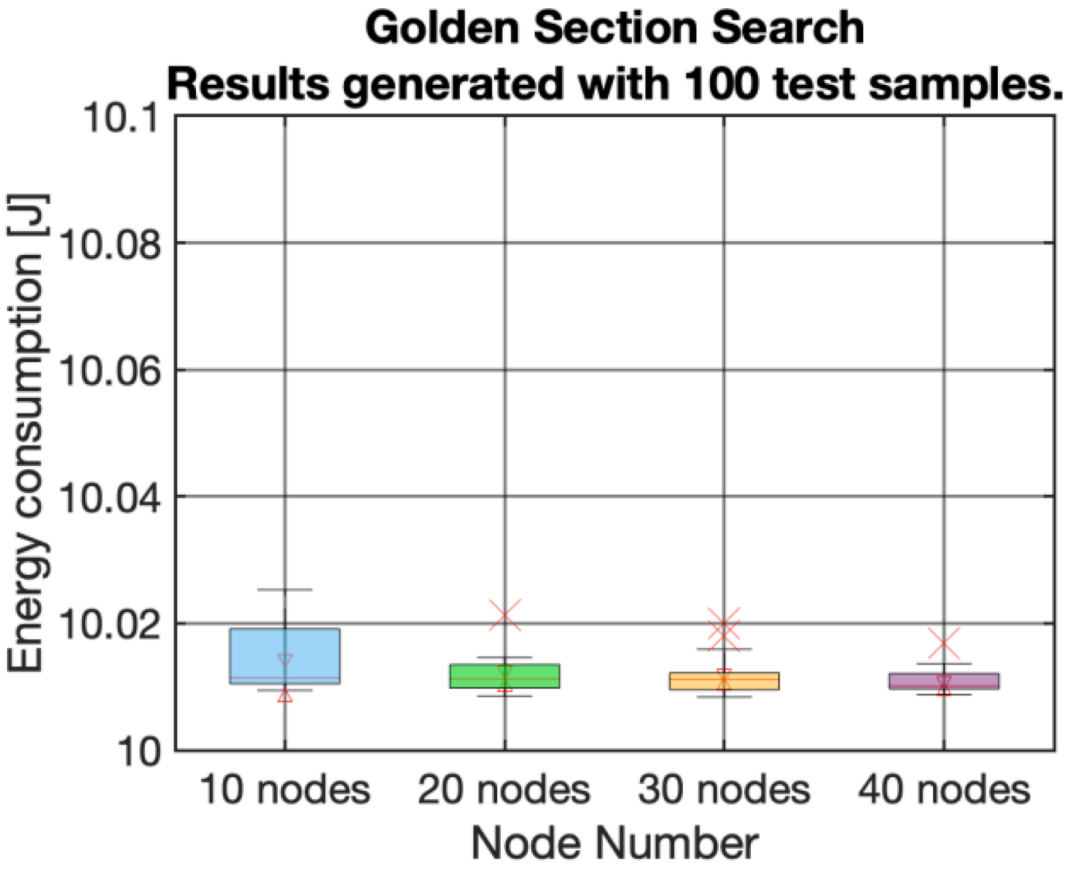

Fig. 8. Results of Golden Section Search algorithm.

time. Heuristic algorithms require less resource than optimization, however they do not always provide optimal solutions, instead they generally produce nearly-optimal results. We defined the problem for using heuristic algorithms as a binary search problem [14] to assign each node to use an encryption algorithm. We define a binary solution vector $S = [S2, S3, ..., S_n]$ that is used as an initial solution for heuristics. Note that, $S_i = 1$ means that node-*i* uses AES and $S_i = 0$ means that node-*i* uses Twofish for encryption. The heuristic algorithms we implemented are Simulated Annealing, Golden Section Search and Genetic algorithms.

6.1. Simulated annealing

Simulated Annealing (SA) algorithm is a probabilistic method for approximating the global optimum. The name comes from the annealing technique in metallurgy, a technique used for heating and supervised cooling of a material to reduce its defects. SA can be used to approximate the global minimum for a function with many variables. In 1983, this method was proposed for solving the traveling salesman problem [25]. SA chooses a random step while searching for the global optimum. If the selected step improves the solution, then it is surely accepted. Otherwise, the random step is accepted with a probability depending on the success of the step. The reason behind accepting a bad move with a given probability is to escape from getting lost in a local optimum. The pseudocode for the algorithm is given in Table 2.

6.2. Golden Section Search

Golden Section Search (GSS) is proposed by Kiefer for computing a minimum/maximum of a unimodal function [25]. Unimodality in mathematics means there is only a sole minimum/maximum point for a defined function. It continuously reduces the solution interval to locate the minimum. Independent of how many points have been evaluated, the minimum value always remains within the interval defined by the two points adjacent to the point with the least value found. The pseudocode for GSS is given in Table 3.

6.3. Genetic algorithm

In computer science, Genetic Algorithm (GA) is used as a meta-heuristic impressed by the means of natural selection. GA is used to find adequate solutions to optimization problems by imitating biological procedures such as mutation, crossover and selection. It initiates by defining a population with a set of solution vectors named chromosomes. Each chromosome in a population is evaluated by considering the objective function. The main idea is searching for more healthy chromosomes within the population (which generate better solutions for the objective function), and then crossover and mutation stages are applied to the chromosomes. Crossover operation is applied to produce new individuals in the population, and mutation operation is involved for escaping the local minima. Chromosomes are represented as binary values of 0s and 1s, which fits well for our binary search problem. The pseudocode for GA is given in Table 4.

We used these three algorithms to make the decision of the encryption algorithm to be used for each node. In this scenario, we forced the networks to use data fragmentation with $L_{node} = 0.5$, selected NSR as 0.8, and as initial solution we have provided the encryption scheme according to Fig. 2. Figs. 7-9 represent the results generated by Simulated Annealing, Golden Section Search and Genetic algorithms respectively. For the analysis of the system, we again ran the algorithms for 100 different randomly generated topologies for 10, 20, 30 and 40 node networks.

Fig. 7 shows the results generated by Simulated Annealing algorithm. It tries to find the optimal energy consumption about using

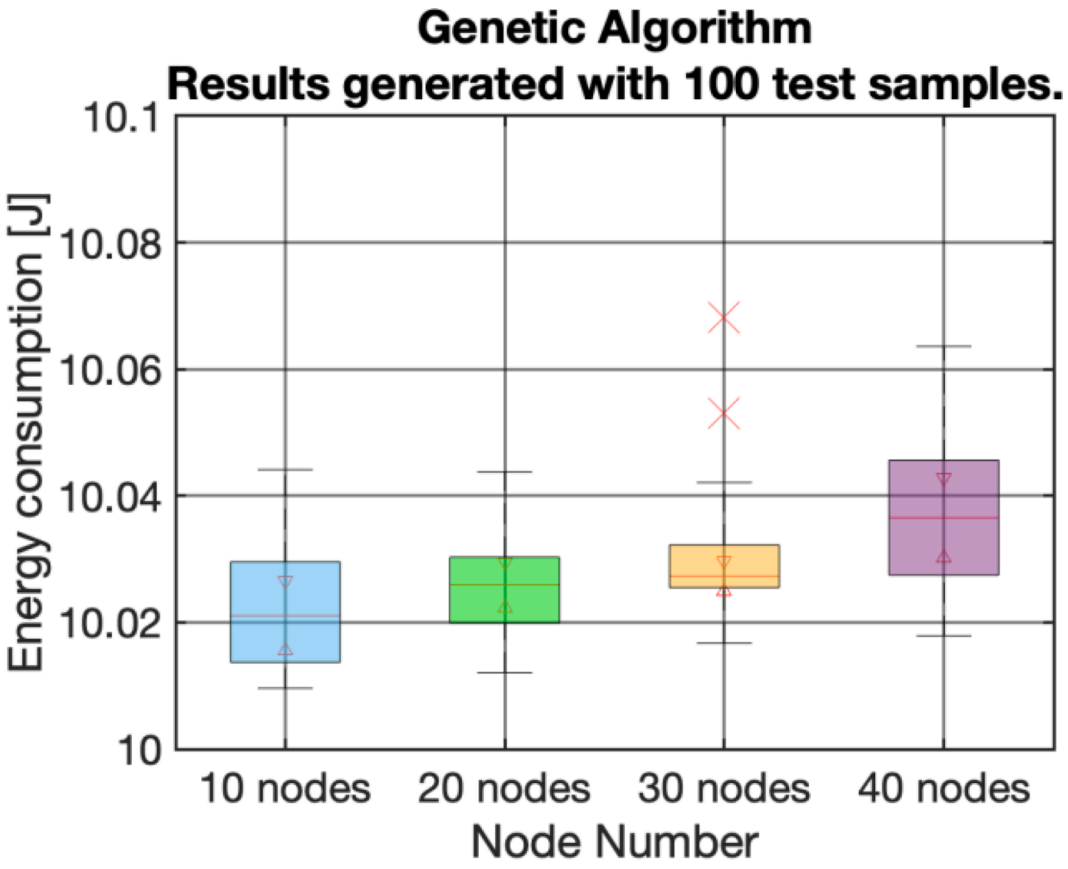

Fig. 9. Results of Genetic Algorithm.

Twofish and AES algorithms for encryption. When we examine the results, we can see that it gives lower results than optimization for using Twofish and AES algorithms together as suggested by our proposed method. We have not limited the algorithm with any constraint about which encryption algorithm to use in the simulations, and as expected, it assigns Twofish algorithm to more nodes opposed to our proposed method, and this is the reason behind producing lower energy consumption results.

Fig. 8 presents the results generated by Golden Section Search algorithm. From the results, we can see that it gives even lower energy consumption results than optimization and Simulated Annealing algorithms for using Twofish and AES encryption together. If we compare the results with optimization results, we understand that Golden Section Search produces nearly close results when we occupy only Twofish as the encryption algorithm, which means that it tends to assign Twofish to every node as the encryption algorithm. Since we did not give any constraints about encryption selection, GSS seems successful about finding minimum energy consumption values. The figure also shows that, GSS works better when the network size is larger.

Fig. 9 reveals the results generated by Genetic Algorithm. From the results, we can see that it gives close energy consumption results to optimization and Simulated Annealing algorithms for using Twofish and AES algorithms together. If we compare the results with optimization results as shown in Fig. 10, we understand that Genetic Algorithm produces nearly close results to our proposed method for deciding encryption algorithm for nodes. This shows that it does not work very efficiently for this problem. From here, we can infer that it is not a very suitable heuristic algorithm for the proposed scenario.

Fig. 10 shows a comparison of optimizations ran with AES method only, Twofish method only, and AES and Twofish method together against the implemented heuristic methods. X-axis represents number of nodes in each network and Y-axis represents the minimum energy spent by the maximum energy consuming node in each network. The results in the Y-axis are found by running simulations for 100 randomly generated networks with 10, 20, 30 and 40 nodes, and then taking their average. From the figure, we can see that using Twofish throughout the network achieves the best energy consumption results, oppositely AES produces the highest energy consumption results. Using AES and Twofish together as proposed in this study limits the increase in the energy consumption, while this strategy maintains an acceptable security level. When we consider the heuristic methods, Golden Section Search comes out as a good solution for finding the nearly-optimum values. Especially for 30 and 40 node networks, the results generated by Golden Section Search is quite close to the optimization results. The important point to stress here is that, when running heuristic methods, we did not give any constraints about which encryption algorithm to choose, hence we expect them to find closer results to Twofish-only optimization results. When viewed from this perspective, Golden Section Search gives the best results among heuristic methods, while Simulated Annealing gives moderate results and Genetic Algorithm gives poor results. On the other hand, Simulated Annealing produces the best result for 10 node networks, which can imply that it might be suitable to use this method for small sized networks. Fig. 10 shows that Genetic Algorithm causes nearly zero change from the initially given solution and this makes it a weak candidate for further usage in UASNs studies.

7. Conclusion

In this paper, an optimal multi-path routing strategy has been developed via mixed-integer programming (MIP) formulations to analyze the effects of multi-path routing, packet duplication, encryption and data fragmentation on network lifetime. The MIP model maximizes the lifetime of the most energy-starving sensor node while it guarantees a pre-determined reliability requirement. In addition, the AES and

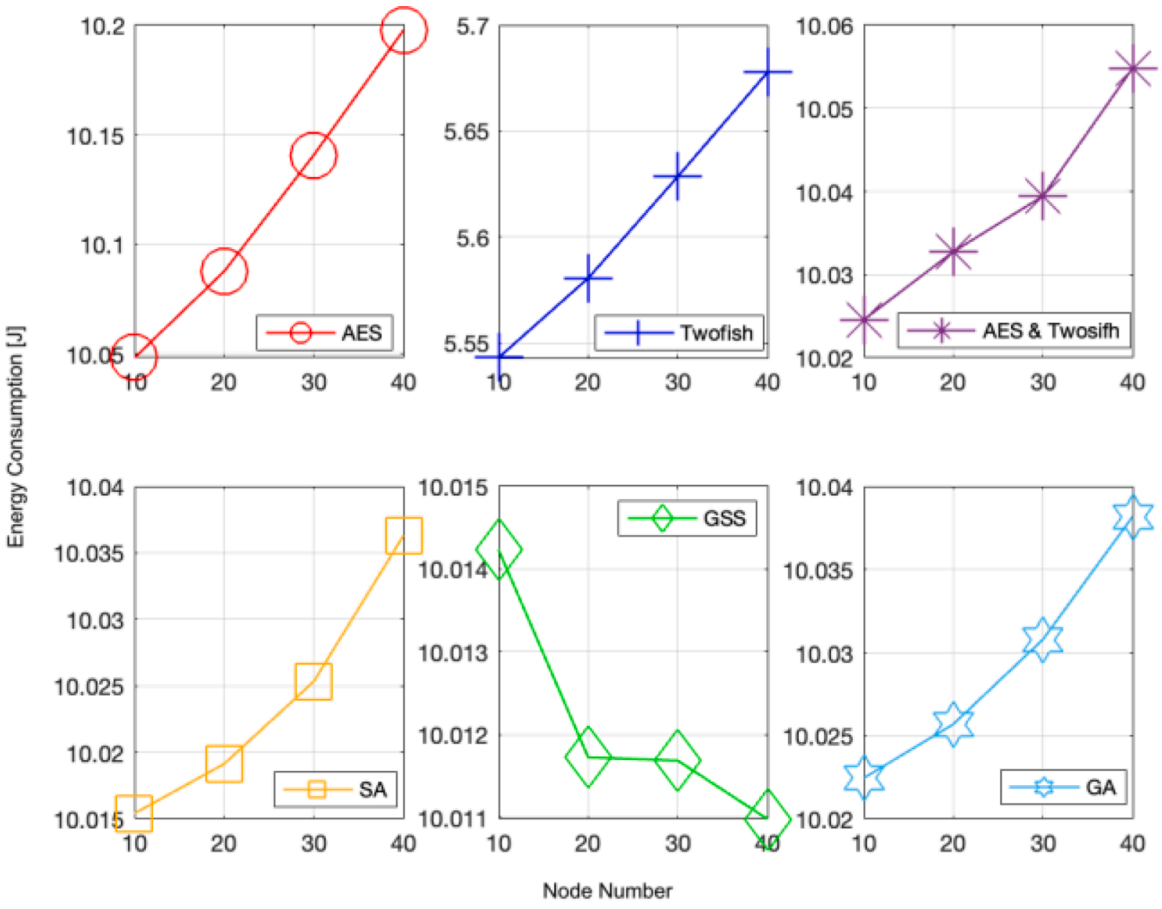

Fig. 10. Comparison of maximum energy consumption of optimization vs. heuristic algorithms.

Twofish encryption algorithms have also been utilized to balance the trade-off between security supplied by encryption and network lifetime in UASNs. Although the proposed optimal multi-path routing strategy is modeled by using an MIP framework, the computational complexity arises from the nature of MIP encourages us to develop meta-heuristic solutions, since meta-heuristic approaches provide sub-optimal solutions in polynomial-time [3–6]. In order to overcome the computational complexity of the optimal multi-path routing strategy which is developed via MIP formulation, we implemented different meta-heuristic approaches, namely, Golden Section Search (GSS), Simulated Annealing (SA), and Genetic Algorithm (GA). For each meta-heuristic algorithm, we investigate the near-optimal solution performance. Numeric performance results of the simulations are listed as follows:

1 In order to maintain a desired network success rate (NSR), we need to sacrifice more energy; thus, more strict reliability rate means more energy consumption for the nodes. When we increase the NSR, energy consumption of the nodes tends to increase. For NSR rates of 0.7, 0.8 and 0.9, the increase in the energy consumption compared to base state (NSR=0.6) is 33%, 77% and 150% respectively.
2 Encryption operation is greedy about energy consumption, however to provide security for the generated data, encryption is a must against eavesdropping attacks. Between the two encryption algorithms employed, using AES for all nodes consumes about 100% more energy compared to using Twofish for all nodes. It is clear that Twofish is better for energy consumption, and it is considered a secure encryption algorithm. Hence, for adequate security we employ Twofish for close nodes to sink and AES for far nodes to sink. In this case, the decrease in the energy consumption against using AES for all nodes is about 20%. By the proposed idea, we can limit the decrease in the network lifetime while maintaining security.
3 Data fragmentation is another promising method against eavesdropping attacks. Nevertheless, it also needs more transmission and reception energy since we are increasing the amount of these operations. Increasing number of data fragments increases the energy consumption in a linear fashion. Forcing the system to use at least 2, 3 and 4 fragments increases energy consumption about 2Joules, 4Joules and 6Joules on average respectively. When we consider the high energy overhead introduced by encryption operation, it is very logical to use data fragmentation jointly to improve the level of the security of the system.
4 When implementing the heuristic methods to decide which encryption algorithm to use for each node, we expected them to find solutions similar to the case where all nodes use Twofish without giving any constraints. For 30 and 40 node networks, GSS finds the approximate results compared to optimization results, which is nearly 3% higher than optimal results. For 20 nodes, GSS still gives the best result that approximates the optimum with 17%. For 10 nodes, SA finds the best result that is nearly 21% higher than optimum value. From these results, we can infer that it is better to use SA for small networks and GSS for larger networks. GA cannot provide good results in the trials, generating only about 1.5% deviated results compared to initially given solution.

CRediT authorship contribution statement

Osman Gokhan Uyan: Conceptualization, Methodology, Software, Writing – original draft, Visualization. **Ayhan Akbas:** Methodology, Software, Writing – review & editing. **Vehbi Cagri Gungor:**

Conceptualization, Writing – review & editing, Supervision, Project administration.

Declaration of Competing Interest

The authors whose names are listed immediately below certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

References

[1] D. Incebacak, K. Bicakci, B. Tavli, Evaluating energy cost of route diversity for security in wireless sensor networks, Comput. Standart. Interfaces (39) (2015) 44–57.

[2] H. Zlatokrilov, H. Levy, Session privacy enhancement by traffic dispersion, in: Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), 2006, pp. 1–12.

[3] J. Cao, J. Dou, Z. Guo, S. Dong, H. Xu, Elt: Energy-level-based hybrid transmission in underwater sensor acoustic networks, in: IEEE 9th International Conference on Mobile Ad-hoc and SensorNetworks, 2013, pp. 133–139.

[4] R. Su, R. Venkatesan and C. Li, "An energy-efficient relay nodeselection scheme for underwater acoustic sensor networks," *Cyber-Physica. Syst.,*1, no. 2-4, pp. 160-179, 2015.

[5] D. Pompili, T. Melodia, I.F. Akyildiz, Routing algorithms for delay-insensitive and delay-sensitive applications in underwater sensor networks, in: Proceedings of the 12th Annual International Conference on Mobile Computing and Networking, ser. MobiCom '06, 2006, pp. 298–309.

[6] J. Chen, X. Wu, G. Chen, Rebar: A reliable and energy balanced routing algorithm for uasns, in: 2008 Seventh International Conference on Grid and Cooperative Computing, 2008, pp. 349–355.

[7] L. Xinbin, C. Wang, Z. Yang, L. Yan, S. Han, Energy-efficient and secure transmission scheme based on chaotic compressive sensing in underwater wireless sensor networks, Digital Signal Process. (81) (2018) 129–137.

[8] C. Castelluccia, E. Mykletun, G. Tsudik, Efficient aggregation of encrypted data in wireless sensor networks, in: The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, San Diego, CA, USA, 2005.

[9] S. Uluagac, R. Beyah, Y. Li, J. Copeland, VEBEK: virtual energy-based encryption and keying for wireless sensor networks, IEEE Trans. Mob. Comput. 9 (7) (2010) 994–1007.

[10] E. Stavrou, A. Pitsillides, A survey on secure multipath routing protocols in WSNs, Comput. Netw. (54) (2010) 2215–2238.

[11] P. Lee, V. Misra, D. Rubenstein, Distributed algorithms for secure multipath routing, in: Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), 2005, pp. 1952–1963.

[12] L. Chen, J. Leneutre, On multipath routing in multihop wireless networks: security, performance, and their tradeoff, Eurasip J. Wireless Commun. Netw. (2009) 1–13.

[13] A. Xenakis, F. Foukalas, G. Stamoulis, T. Khattab, Energy-aware joint power, packet and topology optimization by simulated annealing for wsns, in: 2013 7th IEEE GCC Conference and Exhibition (GCC), 2013, pp. 17–21.

[14] H. Yildiz, K. Bicakci, B. Tavli, H. Gultekin, D. Incebacak, Maximizing wireless sensor network lifetime by communication/computation energy optimization of non-repudiation security service: node level versus network level strategies, Ad Hoc Netw. (37) (2016) 301–323.

[15] E. Hosseini, V. Esmaeelzadeh, M. Eslami, A hierarchical sub- chromosome genetic algorithm (hsc-ga) to optimize power consumption and data communications reliability in wireless sensor networks, Wirel. Pers. Commun. (2) (2015) 752–757.

[16] C. Miller, A. Tucker, R. Zemlin, Integer programming formulation of traveling salesman problems, J. ACM 7 (4) (1960) 326–329.

[17] A. Piltzecker, The Best Damn Windows Server 2008 Book Period, Elsevier, 2011.

[18] J. Daemen and V. Rijmen, "AES proposal: Rijndael", 1999.

[19] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson, The Twofish Encryption Algorithm: A 128-bit Block Cipher, John Wiley & Sons, Inc, 1999.

[20] C. Saifurrab, S. Mirza, M. Tech, AES algorithm using advance key implementation in MATLAB, Int. Res. J. Eng. Technol. (2016) 846–850, v3.4.

[21] I. Vasilescu, K. Kotay, D. Rus, P. Corke, Data collection, storage, and retrieval with an underwater sensor network, in: Proc. Int. Conf. Embed. Netw. Sens. Syst. (SenSys), 2005, pp. 154–165.

[22] A. Akbas, Comparative analysis of lightweight cryptography algorithms on resource constrained microcontrollers, in: 2019 1st International Informatics and Software Engineering Conference (UBMYK), IEEE, 2019, pp. 1–4.

[23] Matlab - mathworks - matlab & simulink, Mathworks (2020) [Online]. Available: https://www.mathworks.com/products/matlab.html.

[24] «IBM Cplex Optimizer,», IBM, 2020 [Online]Available: https://www.ibm.com/analytics/cplex-optimizer.

[25] S. Kirkpatrick, C. Gelatt, M. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.

[26] M. Felemban, E. Felemban, Energy-delay tradeoffs for underwateracoustic sensor networks, in: 2013 First international Black Sea confer-ence on communications and networking (BlackSeaCom), IEEE, 2013, pp. 45–49.

Osman Gokhan Uyan received his B.S degree in Computer Science from Bilkent University, Ankara, Turkey, in 2005. He received his M.S. degree in Electrical and Computer Engineering from Abdullah Gul University, Kayseri, Turkey in 2016 under the supervision of Prof. Dr. Vehbi Cagri Gungor. Currently, he is a Ph.D. student in Electrical and Computer Engineering and Research Assistant at Computer Engineering Department at Abdullah Gul University. His main research interests are in wireless and mobile communications, wireless and ad-hoc sensor networks, fuzzy logic and intelligent optimization algorithms, image processing.

Ayhan Akbas has received B.Sc. and M.Sc degrees from Middle East Technical University, Electrical and Electronics Department in 1991 and 1995 respectively. He earned his Ph.D. in Computer Engineering from TOBB University of Economics and Technology. He worked for multinational companies as SIEMENS, Sun Microsystems, and NEC in technical and managerial positions for over 20 years in the IT business. Currently, he is continuing his career as an academician in Computer Engineering Department at Abdullah Gul University. His areas of interest are Wireless Sensor Networks, Wireless Communication, Computer Networks, IoT, RFID, Mathematical Modelling and Optimizations.

Prof.Dr. V. Cagri Gungor received his B.S. and M.S. degrees in Electrical and Electronics Engineering from METU, Ankara, Turkey, in 2001 and 2003, respectively. He received his Ph.D. degree in electrical and computer engineering from the Broadband and Wireless Networking Laboratory, Georgia Institute of Technology, Atlanta, GA, USA, in 2007. Currently, he is an Associate Professor and Chair of Computer Engineering Department, Abdullah Gul University (AGU), Kayseri, Turkey. His current research interests are in smart grid communications, machine-tomachine communications, next-generation wireless networks, wireless ad hoc and sensor networks, cognitive radio networks. Dr. Gungor has authored several papers in refereed journals and international conference proceedings, and has been serving as an editor, reviewer and program committee member to numerous journals and conferences in these areas. He is also the recipient of TUBITAK Young Scientist Award in 2017, Science Academy Young Scientist Award (BAGEP) in 2017, Turkish Academy of Sciences Distinguished Young Scientist Award (TUBA-GEBIP) in 2014, the IEEE Trans. on Industrial Informatics Best Paper Award in 2012, the European Union FP7 Marie Curie IRG Award in 2009, Turk Telekom Research Grant Awards in 2010 and 2012, and the San-Tez Project Awards supported by Alcatel-Lucent, and the Turkish Ministry of Science, Industry and Technology in 2010.