An Adaptive Confidentiality Security Service Enhancement Protocol Using Image-Based Key Generator for Multi-Agent Ethernet Packet Switched Networks

Jafar J. Abukhait and Ma’en S. Saleh*
Tafila Technical University, Tafila 66110, Jordan
Email: jafar@ttu.edu.jo

***Abstract*—Conventional network Security Protocols provide different types of security services that establish secure connections between network nodes making them robust against different security threats. The security establishment process depends on sharing security keys between connected devices. The level of security provided by such security protocols is static and irrelevant to the status of the network. In this paper, we propose an adaptive security-aware lottery-EDF scheduler for limited resources Ethernet packet switched networks using GAIA multi-agent approach. The proposed system adopts a periodic non-shared cryptographic key-generation process of different sizes (i.e., 64 bits, 128 bits, 160 bits, and 512 bits) using the textural features of digital images stored on a central image-server. The proposed system deploys a security enhancement unit that adaptively enhances the confidentiality security-service level for the real-time packets according to feedback from a congestion control scheme that is based on network resource estimation technique, hence making the network robust against the spoofing security threat while preserving the overall network performance metrics (NPMs). By simulating a real-time Ethernet packet switched network, extensive simulation results show the efficiency of the adaptive proposed protocol over the static-security and IPSec protocols in guaranteeing the QoS requirements for the real-time flows in terms of destination’s buffer consumption and average total packet delays.**

***Index Terms*—Security, confidentiality, key-generator, imaging, QoS, Multi-agents**

I. Introduction

The rapid revolution in the network technologies increases the number of connected nodes from different categories to the internet. In packet switched networks, the network nodes are simultaneously sending real-time data packet flows from different classes that pass through several networks with different technologies till reaching the desired destinations [1]. Such simultaneous sending of real-time flows increases the opportunity for security attackers (threats) to oppose such flows and achieve the desired destructive goals which leads to degrade the level of service provided by the network technologies [2, 3].

To provide the proper service to the real-time data flows and preserve the overall network performance metrics (NPMs), network technologies should be capable of guaranteeing both the quality-of-service (QoS) and the security requirements for the real-time data flows [4, 5]. To guarantee the different QoS metrics (i.e., data delivery, end-to-end delay, jitter, and miss ratio) [6], network technologies deploy several techniques such as traffic prioritization [7], scheduling algorithms [8, 9], resource estimation [10], queuing, and traffic marking [11]. From the other side, applying the appropriate network security service such as confidentiality, authentication, or integrity is the technique followed by network technologies to protect the real-time data flows from different types of security attacks [12, 13]. Such attacks including denial-of-Service, spoofing, snooping, ICMP, SYN floods, sniffing, and peer-to-peer attacks [14].

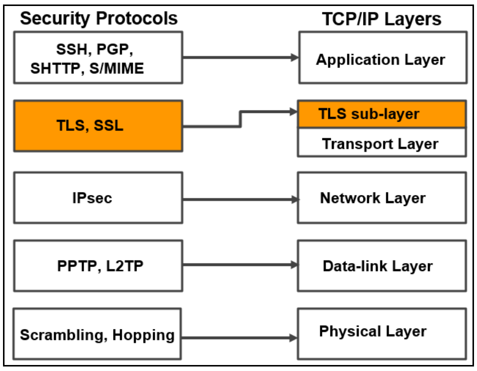


Fig. 1. Security protocols associated with TCP/IP layers.

The previous security services are provided through the implementation of the appropriate security protocol [15]. Different factors play key roles in implementing the security protocol such as the model of the security attack, the network’s topology and available resources, the level of security requested by the network node, network’s technology, and the targeted layer of the TCP/IP model [16]. Accordingly, different security protocols were implemented by the Internet Engineering Task Force (IETF) such as secure socket shell (SSH) [17], secure hypertext transfer protocol (HTTPS) [18], transport later security protocol (TLS), secure socket-layer protocol (SSL) [19], point-to-point tunneling protocol (PPTP), and

Manuscript received September 25, 2022; revised October 25, 2022; accepted October 27, 2022.

*Corresponding author: Ma’en Saleh (email: maen@ttu.edu.jo).

doi: 10.18178/ijeetc.12.2.112-123

Internet protocol security (IPsec) [20]. Fig. 1 shows different types of security protocols by the IETF according to the layer of the TCP/IP model where the security protocol operates.

Internet security association and key management protocol (ISAKM) is required by each security protocol upon initiating a secure session between the interconnected parties. Such protocol provides a framework to exchange security parameters and to exchange security keys (i.e., authentication header (AH) parameters, encapsulation security payloads (ESP) parameters, the security algorithm adopted by the security service, and the predefined key for encryption/decryption processes) [20]. According to the security mechanism followed by the security protocol in initiating a secure secession between the interconnected parties, three main issues should be highlighted: 1) exchanging security parameters and security-key during the security association phase increases the opportunity for security thieves to sniff such parameters and achieve their destructive goals [21]; 2) Once the interconnected parties negotiate on the security parameters and the level of security to be adopted (i.e., the security algorithm) along with the predefined security key, the parameters could not be changed unless the security association phase is pre-initiated (i.e., additional security overhead); 3) The security level adopted by the security protocol is static and is irrelevant to the network's status, that is the security protocol will not adaptively enhances the security level to keep the network performance metrics (NPMs) preserved [22].

In this paper, a GAIA multi-agent system that provides a layer of integration between a security enhancement unit and a congestion control scheme for Ethernet packet switched networks was proposed. The key features of the proposed system are as follows:

- The proposed system eliminates the security association phase between the interconnected parties through implementing a non-shared key generation algorithm. Such elimination decreases the opportunities of altering security parameters and thus the overall security level of the network is enhanced.
- The proposed system adopts a periodic non-shared cryptographic key-generation process of different sizes (i.e., 64 bits, 128 bits, 160 bits, and 512 bits) using the textural features of digital images stored on a central image-server.
- The proposed system deploys a security enhancement unit that adaptively enhances the confidentiality security-service level for the real-time.
- The proposed system preserves the overall network performance metrics (NPMs) through implementing a congestion control mechanism that adopts a resource estimation technique.
- The proposed system was modeled and designed according to the GAIA agent-based methodology, where a cooperation scheme between a set of pre-define sub-agents was implemented.

The rest of the paper is structured as follows: In Section II, we briefly introduce the related work. The multi-agent design model is presented in section III. Section IV illustrated the confidentiality security key generation technique. Section V describes the overall system methodology. Extensive simulations and performance evaluation is fully performed in section VI. Conclusions and future work are finally drawn in section VII.

II. Related Work

Establishing secure sessions between network nodes requires a prior consensus to a key agreement procedure. Accordingly, security key generation process is a key-phase in guaranteeing the security requirements for intercommunicated network nodes. The security-key generation algorithms are following one of three main security key-generation techniques: symmetric, asymmetric and hybrid techniques [23]. Several metrics have been taken into consideration by researchers when proposing the security key generation algorithm in real-time networks such as key length, the class of the security attack, the type of real-time data to be encrypted, the overhead model of the security algorithm, and the overall network performance [24].

In [25], a dynamic symmetric key generation algorithm based on pseudo-random number generator and physical unclonable function was proposed for wireless sensor networks. In [26], a three-dimensional cube symmetric key generation algorithm based on deep neural network learning technique was proposed to provide confidentiality and integrity security services in a quantum computer environment. In [27], an intelligent symmetric security-key generation algorithm based on triple layer vector-valued neural technique was proposed to provide authentication security service for IoT interconnected devices. In [28], an adaptive key-generation technique for Mobile ad-hoc network supporting IoT applications was proposed. The algorithm generates optimal key pairs through the self-adaptive sail fish optimization technique and provides privacy preservation for interconnected nodes. A secure key-establishment algorithm for cluster wireless sensor network was proposed in [29]. The algorithm provides a symmetric key generation technique according to a layered model that take into consideration the limited resources for the WSNs. In [30], a hybrid key establishment scheme that integrates the symmetric and asymmetric cryptography schemes was proposed for WSNs. The hybrid scheme operates in two modes (i.e., symmetric, or asymmetric) according to the level of available resources in the network. The scheme shows high efficiency in providing security aspects to the wireless nodes while keep preserving the overall network resources. In [31], a security key agreement scheme along with rekeying feature was proposed for healthcare applications in WBANs. The proposed scheme shows high efficiency in guaranteeing the confidentiality security service when have been tested in heterogenous WSNs. In [32], an efficient key generation approach based Diffie-Hellman cryptography technique was proposed for packet switched networks. The proposed scheme shows high efficiency in protecting the network from MITMA and performs other security key generation techniques in terms of response time and security overhead.

Researchers implement huge research in generating private security keys from image features. Such algorithms allow the deployment of security schemes that don't share the security key between the interconnected network nodes. Instead, both end parties perform the security key extraction process on a pre-defined image (i.e., colored, gray, binary, or biometric) to generate the non-shared private key needed for the encryption/decryption processes [33]. An image-based security-key generation algorithm in cooperation with SHA-512 was proposed to generate a security key for image encryption by transformation mapping for the generated plaintext related information of an image to the plaintext related security key [34]. In [35], a gray code-based encoding was adopted to generate a pair of public and private keys needed for the RSA authentication algorithm by using two factors: fingerprint biometrics and password. Experimental results show the efficiency of the proposed algorithm in generating stable security keys and preserving the system's resource. In [36], fingerprint biometrics of both the sender and receiver parties were used to generate a non-shared common security key for cryptography. According to the extracted features from the biometric fingerprint, a private permuted binary string is generated that will be used to generate a 256-bit private key using SHA-256 security algorithm. In [37], a security agreement algorithm to generate security keys from medical source images was proposed. The algorithm derives the edge-maps from the source image and plugs them into a three-phase security generation scheme: bit-plane decomposition, random sequence generation, and permutation. Experimental results show high efficiency of the proposed algorithm in generating large key-space that strengthen the encryption process and thus higher protection for the medical images. In [38], a security key generation algorithm for image encryption in cloud computing was proposed. The algorithm generates the security key from the image features extracted by the improved Harris corner optimization and local sensitive hash algorithms. The algorithm shows high efficiency in generating security keys for encryption and reducing the overall security overhead. In [39], a security key generation algorithm from extracted iris features was proposed. The algorithm uses the CNN deep learning model for feature extraction and uses the gradient descent scheme for model training. The algorithm shows high efficiency in improving the encryption/decryption processes by the end parties. In [40], a real-time key generation algorithm using the textural features of hosted images on a central server was proposed. The algorithm generates a non-shared security key with a length of 256 bits to be used by the AES security algorithm in a real-time packet switched network. The overall security model was modeled and designed by agent-based methodology and shows high performance in guaranteeing security and QoS requirements for the real-time flows.

As security and overall network performance trade-off, researchers pay a great attention to implement security-aware models that enhance the security level of the real-time networks while preserving the overall network performance metrics (NPMs) [41, 42]. The main issue in the previous security-aware models is that they don't adaptively modify the security level for the real-time flows according to the high dynamicity in the network's status but instead they adopt the best security level that guarantees the security and the QoS requirements of the real-time flows based on the initial status of the real-time network and thus the overall system performance may be affected. Accordingly, adaptive QoS-aware security models were the best solution to be adopted specially for those limited resources heterogenous networks with both QoS and security requirements [43, 44].

III. Multi-Agent Network Design Model

In this research, we model a real-time Ethernet packet switched network, where N real-time source packet generators are communicating with N real-time destination nodes through secure data sessions. The topology of the real-time network is shown in Fig. 2. The proposed real-time was modeled and designed using GAIA agent-based methodology [42], where both source packet generators and destination nodes are accessing a central image server that provides a database of images needed by such networking nodes to generate the required symmetric private security keys for different confidentiality security algorithms. The destination nodes are part of a local area network (LAN) with a star topology that is connected to a default gateway (edge router) through a layer two interconnecting device (i.e., a hub).

The GAIA agent-based methodology was adopted for our design due to its characteristics that fit our problem as the following: 1) The target is to maximize or minimize some performance metrics in the system (i.e., security level, delay, buffer consumption); 2) Deployed sub-agents are heterogeneous; 3) Interactions between agents are static (i.e., don't change through simulation run-time); 4) number of sub-agents is relatively small.

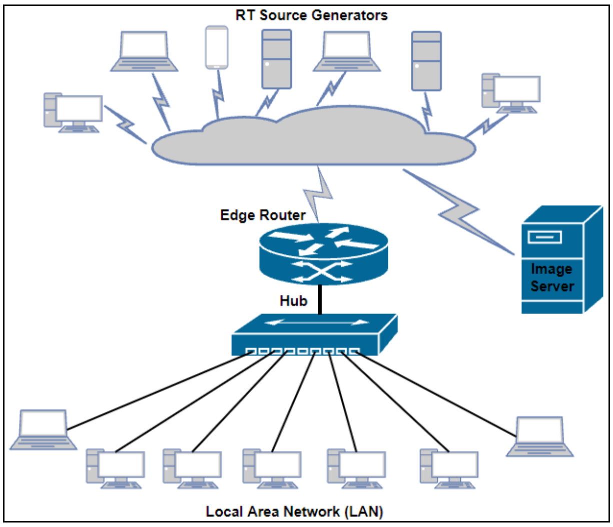


Fig. 2. Network topology.

According to the GAIA agent-based methodology, three design phases were defined: decomposition, modelling, and protocol.

In the decomposition phase, the entire system is decomposed into a set of heterogeneous cooperating agents that are installed at different parties of the real-time network as the following:

- Secure packet generator and key generator agents to be installed at the real-time source node.
- Image selector and source queue (S_Queue) agents to be installed at the image server.
- Security enhancement, router server (R_Server), and router queue (R_Queue) agents to be installed at the edge router.
- Destination server (D_Server) and destination queue (D_Queue) to be installed at the real-time destination node.

In the modelling phase, the main tasks, and behaviors as well as the inputs and outputs for each interactive agent are defined. Finally, the communication protocol that governs the set of interactions between the interactive agents based on well-defined timing constraints is defined in the protocol phase.

A. Secure Packet Generator

This agent is the one responsible for generating a secure real-time flow from one of the following three different classes: video, audio, and best effort (http-text). The security will be achieved by applying the appropriate confidentiality security algorithm on the real-time data flow. The real-time data flow is generated according to the following specifications: 1) a fixed packet length (P_L) of 1500 bytes (i.e. maximum Ethernet frame payload); 2) a flow sending rate (λ); 3) an exponential distribution to model the packet inter-arrival time with a mean (λ^{-1}); 4) a packet deadline (D) that is randomly generated from a uniform distribution over [$D_{\min}$, $D_{\max}$]; 5) An exponential distribution with mean μ^{-1} to model the packet service time, where μ is the packet service rate and is given by:

$$\mu = \frac{\mathrm{Bw}_{\mathrm{agg}}}{8P_L} \tag{1}$$

where $\mathrm{Bw}_{\mathrm{agg}}$ is the aggregate bandwidth for the channel needed for the three types of the data flows (video, audio, and text). Given K data flows in a real-time Ethernet packet switched network, then:

$$\mathrm{Bw}_{\mathrm{agg}} = 0.1K\lambda \tag{2}$$

B. Key Generator

This agent is to be installed at both the source and destination parties in the real-time network. It directly communicates with a centralized image server requesting specific image to be used for generating a list of symmetric private keys with variable lengths. These keys are needed for the encryption/decryption process performed by both source/destination parties, such that the appropriate confidentiality security service is provided for the real-time flow and the whole network is robust against the sniffing (unauthorized access) security threat.

C. Image Selector

This agent is installed at the centralized image server. It is used to retrieve a specific image from the server queue (S_Queue) according to a request from the key generator agent either at source or destination parties. Upon a receiving a request from the source, the image selector uses the request time (at which hour) as an index to be sent to the server queue to retrieve the corresponding image (i.e., if the image server receives a request at time 16:14:32; then the index to be sent is 17). It also keeps a record of each request in a request table that will be used to remove the inconsistency in image selection when serving the requests from the destination side (i.e., the packet received at different time slot than it has been sent). Accordingly, when the destination requests an image, it sends to the image server the information in the received packet (Destination IP address, Source IP address, Destination Port Address, Source Port address). Upon receiving such information, the image selector searches the request table for a matching record and extracts the corresponding image index to be sent to the server queue. The format of the request table is shown in Table I, where both source and destination port numbers are included and thus allowing different application processes at the same source to communicate with different application processes at the same destination (i.e., records 1 and 3).

TABLE I: REQUEST TABLE

Source IP address	Destination IP A address	Source Port address	Destination port address	Image index
S_1	D_1	P_1	P_2	17
S_2	D_2	P_3	P_4	18
S_1	D_2	P_5	P_6	19

D. Server Queue (S_Queue)

This agent en-queues 24 reference images that have been selected from the Kodak Lossless True Color Image Suite [45]. Kodak dataset is used widely as standard test suite for compression testing and visual quality assessment. The 24 reference images are indexed from 1 to 24. Upon receiving a request from the image selector, the S_Queue agent de-queues the image from the queue according to the index number encapsulated in the request and passed it to the image selector without performing any image processing operations on it such as resizing, filtering, rotation, compression, or cropping.

E. Router Server (R_Server)

This agent is to be installed at the edge router. It is responsible of serving the real-time data packets such that the QoS requirements for the real-time data flows are guaranteed. This agent receives the arrived real-time data packets to the router and checks their expiration. If the packet is expired (exceeded the deadline), then it will be dropped and will be recorded in a specific counter (C_{miss}) that monitors the QoS requirements for the real-time flows. If the packet is within the deadline, then the server forwards it to the queue agent to be en-queued according to its class type. This agent adopts a lottery scheduling algorithm to determine the type of the data packet (i.e., video, audio, or text) to be fetched from the queue agent in order to serve it, where a random instance followed a stochastic and sophisticated random number generator distribution is generated. The scheduling algorithm defines a priority vector for the three data classes (video, audio, and text) respectively [P_v, P_a, P_t], such that:

$$\sum_{i=\{a,v,t\}} P_i = 1 \tag{3}$$

F. Router Queue (R_Queue)

This queue at the edge router is divided into three sub-queues to handle the three different classes of the real-time flows (video, audio, and text). Two main processes are performed by such agent: queuing and de-queuing. According to the queuing process, the queue agent stores the unexpired packets arrived from the server agent in the sub-queues based on their type, where each sub-queue has a size Q_x (i.e., $x = \{v, a, t\}$). In each sub-queue, the queuing process follows the EDF algorithm that is the packet with the smallest deadline (closer to expire) will be at the top of the sub-queue (ready to be served). In the de-queuing process, the queue sub-agent responds to a request from the server sub-agent and retrieves a packet from the top of a specific sub-queue and sends it to the server to be served.

G. Security Enhancement

This agent is the core unit in our proposed system. It is the one responsible on adaptively enhancing the confidentiality security service level of the real-time data packets and thus making the real-time network robust against the sniffing security threat. Such security enhancement unit adopts a resource estimation methodology for congestion control according to feedback from the LAN nodes regarding the availability of their resources. Accordingly, while enhancing the security level of the real-time data flows, the overall performance of the network is preserved. In enhancing the security levels of the real-time flows, the security enhancement unit depends on a security overhead model that was based on a study performed on a 175 MHz processor [42], where 8 cryptographic algorithms were applied on a real-time data flow to provide the required confidentiality security service. Table II shows the security overhead model when applying different cryptographic algorithms on the real-time data flow.

TABLE II: CONFIDENTIALITY ALGORITHMS

Index (*J*)	Algorithm	S_j	κ_j
1	SEAL	0.08	168.75
2	RC4	0.14	96.43
3	Blowfish	0.36	37.5
4	Knufu/Khafre	0.40	33.75
5	RC5	0.46	29.35
6	Rijndael	0.64	21.09
7	DES	0.90	15
8	IDEA	1.00	13.5

According to Table II, three metrics were defined for each confidentiality security service algorithm: 1) the strength of the security/algorithm (*J*) with the highest number indicates the strongest algorithm (i.e., IDEA); 2) the processing rate (κ_j) which is the amount of data in KB per unit time (ms) that could be secured using the j^{th} security algorithm; 3) the efficiency metric of the security algorithm relatively to the strongest one (S_j) such that:

$$S_j = 13.5/\kappa_j \tag{4}$$

H. Destination Server (D_Server)

This agent is to be installed at the destination node. Two parameters are defined for such agent: 1) Processing power capability ($\mathcal{P}$); 2) the available memory buffer (*B*). This agent queues the received packets from the edge router and keeps track of its buffer availability. It also interacts with the key-generator agent requesting the required key to decrypt the data packet. This agent responds to periodic requests from the security enhancement agent regarding its resources. Such resources will be used by the security enhancement agent during the process of selecting the optimized security level to be adopted.

IV. GENERATION OF CONFIDENTIALITY SECURITY KEYS

The key generation process has been achieved using gray-level co-occurrence matrix (GLCM) textural features. Since 8 cryptographic algorithms were used in the security enhancement agent, different security key sizes have been deployed. Table III shows the cryptographic algorithms along with their standard key sizes.

TABLE III: CONFIDENTIALITY ALGORITHMS & STANDARD KEY SIZES

Index (*J*)	Algorithm	Key Size/bits
1	SEAL	160
2	RC4	40-2048
3	Blowfish	32-448
4	Knufu/Khafre	512
5	RC5	0-2040
6	Rijndael	128,256,192
7	DES	64
8	IDEA	128

According to Table III, we can generate four different key sizes at the source that are appropriate to these cryptographic algorithms as: 1) 64-bit key for DES algorithm; 2) 160-bit key for SEAL algorithm; 3) 512-bit key for Knufu/Khafre algorithm; and 4) 128-bit key for the remaining algorithms. The key generation process is adaptive and follows three phases as shown in Fig. 3.

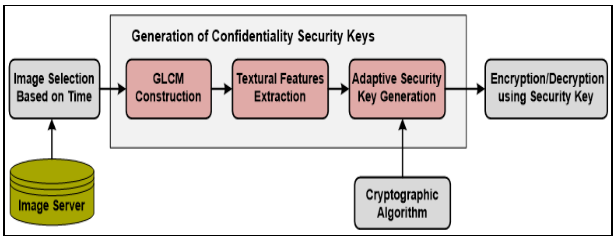

Generation of Confidentiality Security Keys
Image Selection Based on Time
GLCM Construction
Textural Features Extraction
Adaptive Security Key Generation
Encryption/Decryption using Security Key
Image Server
Cryptographic Algorithm

Fig. 3. Phases of confidentiality security keys generation.

A. Secure Packet Gray Level Co-occurrence Matrix (GLCM) Construction

The gray level co-occurrence matrix (GLCM) is the distribution of co-occurring gray levels at specific direction and offset. It measures the frequencies of either horizontal or vertical neighboring pixels [46]. This step generates two 8×8 co-occurrence matrices (P_{ij}) for horizontal and vertical neighboring pixels denoted by *i* and *j* respectively; after quantizing the reference images to eight gray levels. The horizontal matrix would be used to generate the security keys for all cryptographic algorithms where the vertical one would be used only for Knufu/Khafre cryptographic algorithm.

B. Textural Features Extraction

This step extracts the following textural features from the GLCM matrix: Uniformity, Contrast, Correlation, Homogeneity, Entropy, Autocorrelation, Dissimilarity, and Cluster Shade [47]. Eqs. (5) to (8) define a set of these features given that: $p(i,j)$ is the $(i,j)^{th}$ element in the co-occurrence matrix, μ_x and μ_y are the mean values for both the rows and columns of the matrix, σ_x and σ_y are the standard deviation values for both the rows and columns of the matrix, and N_g is the number of gray levels in the scaled image [46–48]:

$$\mu_x = \sum_i \sum_j ip(i,j) \tag{5}$$

$$\mu_y = \sum_i \sum_j jp(i,j) \tag{6}$$

$$\sigma_x = \sum_i \sum_j (i-\mu_x)^2 \, p(i,j) \tag{7}$$

$$\sigma_y = \sum_i \sum_j (j-\mu_y)^2 \, p(i,j) \tag{8}$$

The textural features are defined as shown in Eqs. (9-16) [46-48]:

1) Uniformity (UNI) given as:

$$\text{UNI} = \sum_i \sum_j p(i,j)^2 \tag{9}$$

2) Entropy (ENT) given as:

$$\text{ENT} = -\sum_i \sum_j p(i,j)\log\left(p(i,j)\right) \tag{10}$$

3) Dissimilarity (DIS) given as:

$$\text{DIS} = \sum_i \sum_j |\, i-j \,|\, p(i,j) \tag{11}$$

4) Contrast (CON) given as:

$$\text{CON} = \sum_{n=0}^{N_{g-1}} n^2 \left\{ \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \right\} \tag{12}$$

5) Inverse Difference (INV) given as:

$$\text{INV} = \sum_i \sum_j \frac{p(i,j)}{1+|\, i-j \,|} \tag{13}$$

6) Inverse Difference Moment (IDM) given as:

$$\text{IDM} = \sum_i \sum_j \frac{p(i,j)}{1+(i-j)^2} \tag{14}$$

7) Cluster Shade (CLS) given as:

$$\text{CLS} = \sum_i \sum_j \left(i+j-\mu_x-\mu_y\right)^3 p(i,j) \tag{15}$$

8) Cluster Prominence (CLP) given as:

$$\text{CLP} = \sum_i \sum_j \left(i+j-\mu_x-\mu_y\right)^4 p(i,j) \tag{16}$$

C. Adaptive Security Key Generation

Each textural feature value is converted from its single-precision floating-point format to a 32-bit binary code using the IEEE 754 converter [49]. A specific number of these 32-bit binary codes are concatenated to form the required security key according to the cryptographic algorithm. Four security keys with different sizes are generated as follows:

- 64-bit key for DES algorithm using the Uniformity and Entropy features.
- 160-bit key for SEAL algorithm using the Uniformity, Entropy, Dissimilarity, Contrast and Inverse Difference features.
- 512-bit key for Knufu/Khafre algorithm using the eight textural features from both the horizontal and vertical neighboring GLCM to obtain a 16 feature.
- 128-bit key for the remaining algorithms using the Uniformity, Entropy, Dissimilarity and Contrast.

V. System Methodology

The proposed agent-based system adaptively enhances the confidentiality security service level of the real-time flows needed to combat the sniffing security threat, while keep preserving the network performance metrics (NPMs) for the real-time Ethernet packet switched network.

According to the agent-based model, the process begins when a source node has a real-time flow that is needed to be secure against the sniffing security threat. Accordingly, the key generator agent at the source node interacts with the image selector agent at the image server (i.e., the destination IP address is the image server) requesting for an image. Upon receiving such request, the image selector extracts the request time (at which hour) and sends it to the server queue agent. It also keeps a record for each request as in Table I. The server queue uses the received time information from the image selector as an index to its image database to retrieve the required image accordingly and passes it to the image selector. The image selector then sends the retrieved image to the key generator agent at the source node (i.e., the destination IP address is the node IP address). Upon receiving such image, the image selector generates the security keys for the confidentiality security service algorithms using the technique showed in section 4.

The secure packet generator starts generating its secure real-time data packets (video, audio, or text) with well-defined flow specifications. Initially, it adopts a mid-level security service algorithm (i.e., Knufu/Khafre) till be notified by the security enhancement agent at the edge router with the optimal security service level to be adopted such that the real-time network will not be congested by heavy traffic loads.

In order to preserve the NPMs for the Ethernet packet switched network and provides a higher level of security against security threats, our proposed model adopts a non-shared cryptographic key generation process at both sender and receiver, thus eliminating the security association overhead performed by the IPsec protocol to exchange the security keys. In implementation, the proposed protocol overloads the 3-bit priority code point (PCP) field of the IEEE 802.1Q frame format to represent one of the eight confidentiality security algorithms that was applied on the real-time data packets (i.e., SEAL {000} - IDEA {111}) as shown in Fig. 4. Accordingly, the destination node checks this field of the arrived packet to determine the applied security algorithm by the source node and uses the appropriate security key to decrypt such secure packet [42].

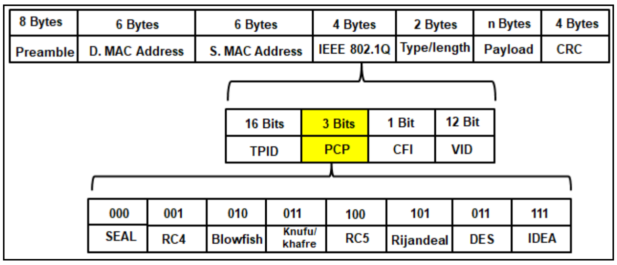


Fig. 4. Overloading PCP field of IEEE 802.1Q.

Upon receiving a real-time data packet to the edge router, the R_Server agent checks its deadline. If the packet is expired, the R_Server drops it and increments the C_{miss} counter that's corresponding to a specific destination. Otherwise, it passes it to the R_Queue. Accordingly, the R_Queue checks the type of such arrived packet from the R_Server and en-queues it in the corresponding sub-queues (video, audio, or text) following the EDF scheduling algorithm (i.e., the closet to expire is at the top of the sub-queue). When the R_Server agent is idle (i.e., completes serving a packet), it interacts with the R_Queue agent requesting for a packet from a specific sub-queue (i.e., the one with the highest priority according to the lottery algorithm performed by the R_Server). The R_Queue responds by fetching the packet from the top of the selected sub-queue. Upon receiving such packet, the R_Server checks its deadline. If the packet is expired, the R_Server drops it and increments the corresponding C_{miss}. Otherwise, it serves the packet and passes it to the destination.

To decrypt the arrived secure real-time data packet at the destination, the D_Server checks the PCP field of the packet and sends its value to the key generator agent. The key-generator responds by the sending the appropriate security key corresponding to such PCP value from a pool of security keys (i.e., for each confidentiality security algorithm) that have been generated following the procedure in Section 4. While serving the arrived data packets, the D_Server keeps track of two main parameters for each traffic flow sent by a specific source generator: 1) the number of arrived data packets (m); 2) the summation of time periods between successive arrived data packets (ω) such that:

$$\omega = \sum_{i=2}^{m} \left(t_i - t_{i-1} \right) \tag{17}$$

To adaptively enhancing the security level of the real-time traffic flow such that the overall network will not be congested by heavy traffic loads, the security enhancement agent keeps track of the destination nodes resources periodically. Over every time period (T), the security enhancement agent broadcasts a control message in the LAN requesting the destination nodes for their resources (i.e., ζ and B) and system parameters (m and ω). For a specific destination node receiving a real-time traffic from a specific source generator, the security enhancement evaluates the mean-inter-arrival time for the arrived packets at the destination buffer (Ψ):

$$\Psi = \frac{m-1}{\omega} \tag{18}$$

In order to find the best confidentiality security level (h) to be adopted by the source generator in securing the data packets, the security enhancement uses the security overhead model in Table II that is based on a study performed on a 175 MHz processing machine to find the memory resources (M) needed to enhance a total of ω real-time data packets with a fixed size of 1500 bytes using each confidentiality security service level (j) that is:

$$M_j = \omega \frac{1500\text{Bytes} \times 175\text{MHz}}{\kappa_j \zeta} \tag{19}$$

Accordingly, the best confidentiality security level (h) to be adopted is chosen such that:

$$M_h \le B < M_{h+1} \tag{20}$$

The security enhancement agent interacts with the source generator agent notifying it with the decision on the security level (h) to be adopted. Accordingly, the source agent might enhance the security level to a higher level, a lower level or stay at the same security level (i.e., the decision on the security level is the same as the one used by the source generator). Fig. 5 shows a diagram of the agent-based model for the real-time Ethernet packet switched network along with the interactions between system agents.

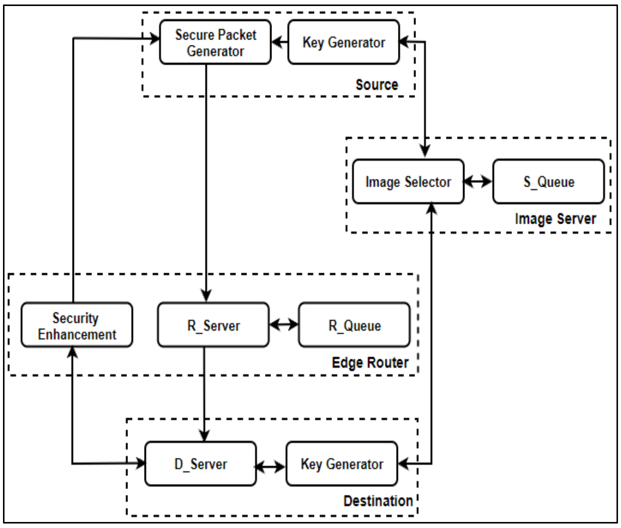


Fig. 5. Agent-based model.

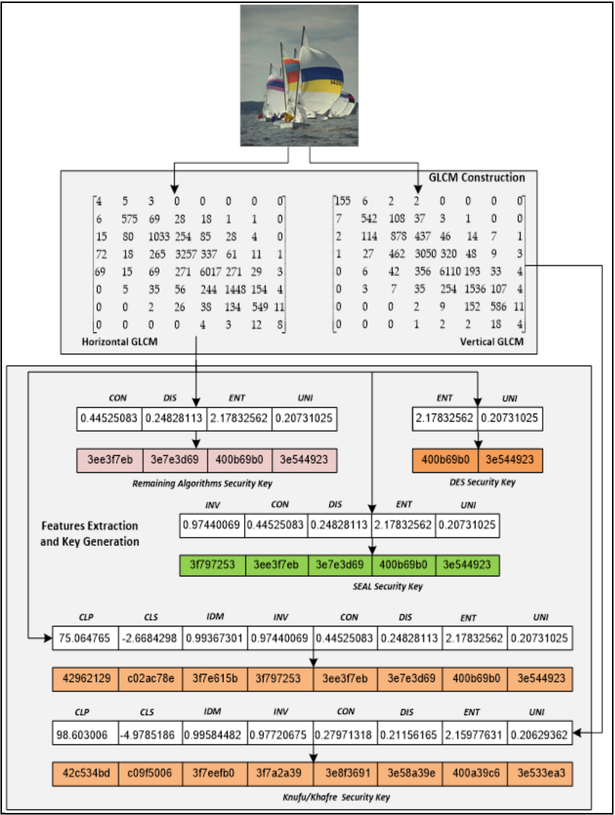
VI. Simulation and Results

To measure the performance of our proposed system, we simulate a real-time Ethernet packet switched network with K pairs of distinct source/destination real-time nodes. Although redesigning the Ethernet frame format for security issues limits the process of measuring and testing the NPMs in real world systems, the system parameters used to run the simulation process are generated through experiments performed on real-time Ethernet packet switched network (i.e., Table II). In the following sections, we will analyze the performance of the proposed algorithm from different perspectives.

A. Confidentiality Security keys Generation

Twenty-four gray scale images were hosted on the image server where each one has an integer label from 1 to 24. These images have been obtained from the Kodak Lossless True Color Image Suite [45]. Each image has a scale of 768×512 or 512×768 pixels. The image selector agent would use these labels to retrieve the desired image based on time once it received a request from either the source or destination. Fig. 6 shows sample of three images on the image server.

Fig. 6. Sample of images hosted on the image server.


GLCM Construction
Horizontal GLCM
Vertical GLCM
Features Extraction and Key Generation
Remaining Algorithms Security Key
DES Security Key
SEAL Security Key
Knufu/Khafre Security Key

$$\begin{bmatrix} 4 & 5 & 3 & 0 & 0 & 0 & 0 & 0 \\ 6 & 575 & 69 & 28 & 18 & 1 & 1 & 0 \\ 15 & 80 & 1033 & 254 & 85 & 28 & 4 & 0 \\ 72 & 18 & 265 & 3257 & 337 & 61 & 11 & 1 \\ 69 & 15 & 69 & 271 & 6017 & 271 & 29 & 3 \\ 0 & 5 & 35 & 56 & 244 & 1448 & 154 & 4 \\ 0 & 0 & 2 & 26 & 38 & 134 & 549 & 11 \\ 0 & 0 & 0 & 0 & 4 & 3 & 12 & 8 \end{bmatrix} \quad \begin{bmatrix} 155 & 6 & 2 & 2 & 0 & 0 & 0 & 0 \\ 7 & 542 & 108 & 37 & 3 & 1 & 0 & 0 \\ 2 & 114 & 878 & 437 & 46 & 14 & 7 & 1 \\ 1 & 27 & 462 & 3050 & 320 & 48 & 9 & 3 \\ 0 & 6 & 42 & 356 & 6110 & 193 & 33 & 4 \\ 0 & 3 & 7 & 35 & 254 & 1536 & 107 & 4 \\ 0 & 0 & 0 & 2 & 9 & 152 & 586 & 11 \\ 0 & 0 & 0 & 1 & 2 & 2 & 18 & 4 \end{bmatrix}$$

CON	DIS	ENT	UNI
0.44525083	0.24828113	2.17832562	0.20731025
3ee3f7eb	3e7e3d69	400b69b0	3e544923

ENT	UNI
2.17832562	0.20731025
400b69b0	3e544923

INV	CON	DIS	ENT	UNI
0.97440069	0.44525083	0.24828113	2.17832562	0.20731025
3f797253	3ee3f7eb	3e7e3d69	400b69b0	3e544923

CLP	CLS	IDM	INV	CON	DIS	ENT	UNI
75.064765	-2.6684298	0.99367301	0.97440069	0.44525083	0.24828113	2.17832562	0.20731025
42962129	c02ac78e	3f7e615b	3f797253	3ee3f7eb	3e7e3d69	400b69b0	3e544923

CLP	CLS	IDM	INV	CON	DIS	ENT	UNI
98.603006	-4.9785186	0.99584482	0.97720675	0.27971318	0.21156165	2.15977631	0.20629362
42c534bd	c09f5006	3f7eefb0	3f7a2a39	3e8f3691	3e58a39e	400a39c6	3e533ea3

Fig. 7. Key Generation phases on a sample image.

The key generator agent on the source or destination follows three steps to generate the confidentiality security key with the required length. The first step is constructing two 8×8 GLCM matrices of horizontally and vertically neighboring pixels after scaling the image to eight gray levels. The second step is extracting eight different textural features from these GLCM matrices where each feature value is single-precision floating-point number. In the third step, a 32-bit binary code (hexadecimal code) is obtained from each feature value using the IEEE 754 convertor. Table IV and Table V show the eight feature values and hexadecimal codes of images I_7 and I_{21}. Then, these binary codes are concatenated each time according to the confidentiality algorithm to generate a security key of the required length. Fig. 7 demonstrates an example of key generation on one image.

TABLE IV: FEATURE VALUES & CODES FOR IMAGE I_7

Feature	Image I7			
	Horizontal GLCM		Vertical GLCM	
	Value	Code	Value	Code
UNI	0.1552912	3E1F04A7	0.1444688	3E13EF9F
ENT	2.2943313	4012D653	2.3887558	4018E160
DIS	0.2859953	3E926DFA	0.3838172	3EC483AF
CON	0.3733105	3EBF228D	0.5453909	3F0B9EBD
INV	0.9691610	3F781AEF	0.9590345	3F758349
IDM	0.9944131	3F7E91DB	0.9919725	3F7DF1E8
CLS	2.7835284	C0322554	1.2356876	BF9E2B03
CLP	61.797199	42773055	47.454350	423DD141

TABLE V: FEATURE VALUES & CODES FOR IMAGE I_{21}

Feature	**Image I_{21}**			
	Horizontal GLCM		Vertical GLCM	
	Value	Code	Value	Code
UNI	0.1552912	3e1f04a7	0.1896470	3e4232d4
ENT	2.2943313	4012d653	2.1883644	400c0e29
DIS	0.2859953	3e926dfa	0.2180976	3e5f54fa
CON	0.3733105	3ebf228d	0.2890412	3e93fd35
INV	0.9691610	3f781aef	0.9765158	3f79fcf1
IDM	0.9944131	3f7e91db	0.9956980	3f7ee611
CLS	2.7835284	c0322554	0.9108296	3f692c21
CLP	61.797199	42773055	130.87476	4302dff0

B. System Parameters & Assumptions

To run our extensive simulations and measure the network performance metrics of our proposed system from different perspectives, we assign the required system parameters and assumptions as shown in Table VI.

TABLE VI: SYSTEM PARAMETERS & ASSUMPTIONS

Parameter	**Value**
Number of node pairs (K)	10, 20, 30, 40, 50, 60
Deadline for video streams	[50 ms, 160 ms]
Deadline for audio streams	[175 ms, 245 ms]
Deadline for text streams	[260 ms, 340 ms]
Packet length P_L	1500 Bytes
Sending rate (λ)	2500 packets/sec.
Initial Security Level (J)	4 (Knufu/Khafre)
Flow mean-inter arrival time (λ-1)	1/2500
Initial Destination buffer (B)	λ /20, λ /10, or unbounded
Edge router sub-queue size (Qx)	Unbounded
Traffic priority vector [Pv, Pa, Pt]	[0.6, 0.3, 0.1]
Node Processing power (ζ)	2 GHz
Time period (T)	30 ms
Initial aggregate bandwidth	0.1*10*2500 =2500 (equ.2)
Initial packet service rate (μ)	2500/(8*1500) =0.208 (equ.1)
Number of Images in the database	24
Image scales	768×512 or 512×768 Pixels

C. QoS at the Destination Agent

In this simulation, we measure the performance of the proposed system at the destination agent using two QoS metrics: 1) the average total packet delays; 2) the buffer consumption. The adaptive proposed protocol was compared with two other protocols: 1) the IPSec protocol, where the system is adaptively enhances the confidentiality security levels of the packets with a security association process between the two parties is established each time a decision on the security level was made; 2) static security enhancement protocol, where the system is using the IPSec. Protocol with the steady state security level that we achieved by the adaptive proposed protocol all the time (i.e., no congestion control is implemented).

Fig. 8 shows the average steady state confidentiality security level achieved by the proposed system when simulating real-time Ethernet packet switched networks with variable number of network nodes (i.e., K= 10, 20, 30, 40, 50, 60) for different initial destination memory buffer (B) (i.e., B= {λ_f /20, λ_f /10, and unbounded}).

As we can see from the simulation result, the confidentiality security level will go higher for destinations with larger initial buffer length (P_L), where the overhead of decrypting high security level packets will not overflow the destination buffer by the new arrived data packets. The result also shows that the average steady state confidentiality security level will go higher for networks with high node density, where the edge router schedules packets for more destination nodes and thus the arrival rates will be lower as shown in Table VII. Such low arrival rate gives the destination node more flexibility in processing high security level packets.

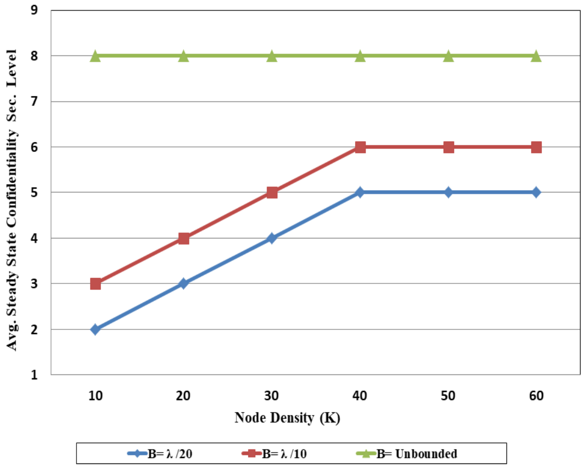


Fig. 8. Average steady state confidentiality sec. Level

TABLE IVII: ARRIVAL RATE AT THE DESTINATION AGENT

Node density (K)	10	20	30	40	50	60
Arrival rate (Packets/sec.)	2348	2308	2287	2265	2262	2260

According to the steady state confidentiality security level values obtained in Fig. 8, we measure the performance of the adaptive proposed system at the destination agent for the two previously mentioned QoS metrics (the buffer consumption and the average total packet delays). We have used an initial buffer size (B) to be (λ_f /10) and thus, the steady state security level to be used by the static security enhancement protocol is 6 as shown from the results in Fig. 8. As shown from Fig. 9, the proposed adaptive protocol.

Fig. 10 shows the tradeoff between the higher security level achieved and the QoS at the destination in terms of average total packet delays. However, the simulation result shows that our adaptive protocol outperforms the other protocols in such metric. Such enhancement is due to the optimized buffer utilization of the destination buffer through the implementation of the congestion control mechanism, and thus the waiting time for the packets at the destination buffer will be minimized (less average total packet delay).

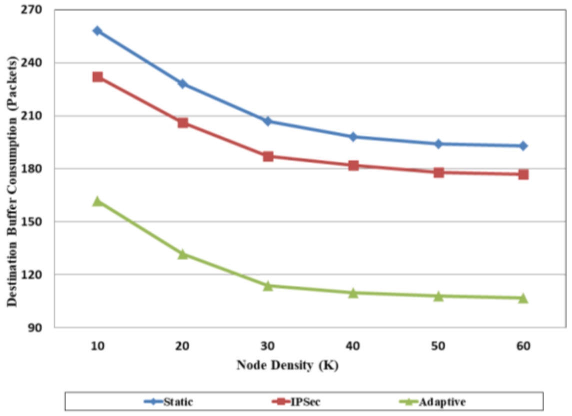


Fig. 9. Average destination buffer consumption (B= λ_f /10).

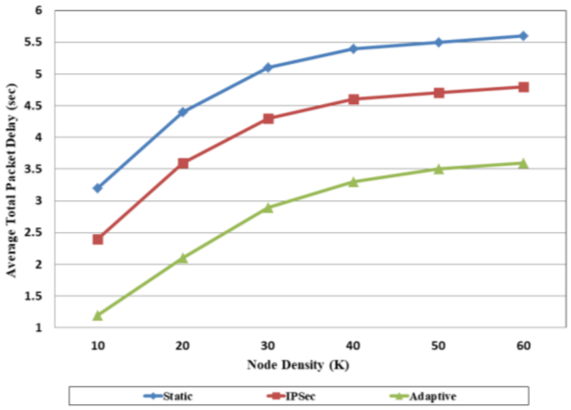


Fig. 10. Average total packet delays at destination (B= λ_f /10)

Although the IPSec protocol implemented here is an adaptive one, but the QoS metrics at the destination agent (i.e., buffer consumption and the average total packet delay) are affected by the overhead of the security association process performed by the two parties (source/destination) on every security level update, where the secure queued packets at the destination will be delayed till the security association process is accomplished. As a result, more buffer consumption due to the new arrived data packets and more average total packet delay.

The IPSec security association process is accomplished through two main phases: 1) the secure session establishment; 2) the exchanging of security parameters needed to complete the security enhancement process (i.e., shared symmetric private key and the security algorithm). According to a real-time experiment performed on 206 MHz network nodes, the results show that the overhead of the security association process is 167 msec. Since our proposed system adopts a non-shared cryptographic key generation process through overloading the PCP field of the IEEE802.1Q, then the total overhead of the security associations (Φ) that will be eliminated by our proposed is system is given as:

$$\Phi = V\left(\frac{\zeta}{206\text{MHz}}\right)167\text{ms} \tag{20}$$

where V is the number of security level changes during the security enhancement process. Fig. 11 shows the average number of security level changes when applying the IPsec protocol with a congestion control mechanism and which have been eliminated by our proposed protocol.

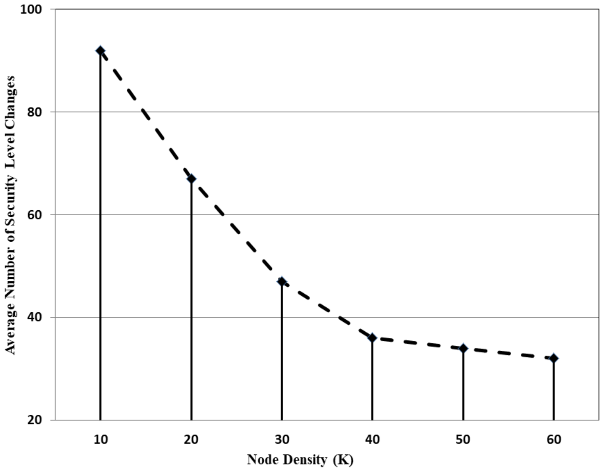


Fig. 11. Average Num. of security level changes (B= λ_f/10; T=30 ms).

VII. Conclusion

In this research, an adaptive multi-agent model for confidentiality security service enhancement in Ethernet packet switched network was proposed. The proposed system performs an integration between a congestion control unit that adopts a resource estimation scheme and a security enhancement unit. The proposed system implements a security-key generation mechanism that uses the extracted textural features of digital images to generate multiple lengths private keys for different confidentiality service algorithms. Compared to other security protocols, the proposed system shows high efficiency in guaranteeing the QoS and security requirements of the real-time data flows and preserving the overall NPMs. The future work of this research includes enhancing the security levels for other security services (i.e., integrity and authentication) to combat other different classes of security threats. It also includes implementing a new biometric security key generation based on deep learning algorithms.

Conflict of Interest

The author declares no conflict of interest.

Author Contributions

Jafar J. Abukhait collected image-based data, carried out the simulation related to the process of generating the confidentiality security keys, participated in analyzing simulation results and drafting the manuscript. Ma'en S. Saleh carried out the multi-agent design model, implemented the integrated agent-based simulation model, and participated in analyzing simulation results and drafting the manuscript. Both authors read and approved the final manuscript.

References

[1] M. Chiesa, A. Kamisiński, J. Rak, G. Rétvári, and S. Schmid, "A survey of fast-recovery mechanisms in packet-switched networks," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1253-1301, 2021.

[2] N. Y. Saigushev, U. V. Mikhailova, O. A. Vedeneeva, and A. A. Tsaran. "Information systems at enterprise: Design of secure network of enterprise," *Journal of Physics: Conf. Series*, vol. 1015, no. 4, 2018.

[3] P. Kemparaj and S. S. Kumar, "Secure precision time protocol in packet switched networks," in *Proc. IEEE Int. Symp. on Precision Clock Synchronization for Measurement, Control, and Communication*, 2019.

[4] K. Sood, K. K. Karmakar, S. Yu, V. Varadharajan, S. R. Pokhrel, and Y. Xiang, "Alleviating heterogeneity in SDN-IoT networks to maintain QoS and enhance security," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5964-5975, July 2020.

[5] H. Babbar, S. Rani, S. M. N. Islam, and S. Iyer, "QoS based security architecture for software-defined wireless sensor networking," in *Proc. 6th Int. Conf. on Innovative Technology in Intelligent System and Industrial Applications (CITISIA)*, 2021.

[6] E. Gelenbe, J. Domanska, P. Fröhlich, M. P. Nowak, and S. Nowak, "Self-aware networks that optimize security, QoS, and energy," *Proceedings of the IEEE*, vol. 108, no. 7, pp. 1150-1167, July 2020.

[7] F. T. Zuhra, *et al*., "LLTP-QoS: Low latency traffic prioritization and QoS-aware routing in wireless body sensor networks," *IEEE Access*, vol. 7, pp. 152777-152787, 2019

[8] M. E. Saleh, "Secure scheduling for wireless networks using NxN wireless switch," *Int. Journal of Electrical, Electronics & Computer Systems*, vol. 18, no. 2, pp. 1-8, 2014.

[9] M. Saleh, A. Aljaafreh, and N. Al-Oudat, "Hierarchal scheduling algorithm for congestion traffic control using multi-agent systems," *Int. Journal of Advanced Computer Research*, vol. 4, no. 4, pp. 915-921, 2014.

[10] M. Saleh, "Adaptive security-aware cone-shaped request-zone location-aided routing protocol using agent-based methodology for VANETs," *Journal of Communications*, vol. 17, no. 5, pp. 308-321, 2022.

[11] F. Fejes, S. Nádas, G. Gombos, and S. Laki, "DeepQoS: Core-stateless hierarchical QoS in programmable switches," *IEEE Trans. on Network and Service Management*, vol. 19, no. 2, pp. 1842-1861, 2022.

[12] M. Bartłomiejczyk, I. E. Fray, M. Kurkowski, S. Szymoniak, and O. Siedlecka-Lamch, "User authentication protocol based on the location factor for a mobile environment," *IEEE Access*, vol. 10, pp. 16439-16455, 2022.

[13] A. N. Rukavitsyn, K. A. Borisenko, I. I. Holod, and A. V. Shorov, "The method of ensuring confidentiality and integrity data in cloud computing," in *Proc. XX IEEE Int. Conf. on Soft Computing and Measurements (SCM)*, 2017, pp. 272-274.

[14] J. Hou and X. Jia, "Research on enterprise network security system," in *Proc. 2nd Int. Conf. on Computer Science and Management Technology (ICCSMT)*, 2021, pp. 216-219.

[15] S. Braun, C. T. Cheng, S. Dowey, and J. Wollert, "Survey on security concepts to adapt flexible manufacturing and operations management based upon multi-agent systems," in *Proc. IEEE 29th Int. Symposium on Industrial Electronics (ISIE)*, 2020, pp. 480-484.

[16] S. Ponmaniraj, R. Rashmi, and M. V. Anand, "IDS based network security architecture with TCP/IP parameters using machine learning," in *Proc. Int. Conf. on Computing, Power and Communication Technologies (GUCON)*, 2018, pp. 111-114.

[17] R. Andrews, D. A. Hahn, and A. G. Bardas, "Measuring the prevalence of the password authentication vulnerability in SSH," in *Proc. IEEE Int. Conf. on Communications (ICC)*, 2020.

[18] R. Shah and S. Correia, "Encryption of data over HTTP (hypertext transfer protocol)/HTTPS (hypertext transfer protocol secure) requests for secure data transfers over the internet," in *Proc. Int. Conf. on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, 2021, pp. 587-590.

[19] A. Liu, A. Alqazzaz, H. Ming and B. Dharmalingam, "Iotverif: automatic verification of SSL/TLS certificate for IoT applications," *IEEE Access*, vol. 9, pp. 27038-27050, 2021.

[20] "IEEE Standard for Interoperability of Internet Protocol Security (IPsec) Utilized within Utility Control Systems," *IEEE Std 2030.102.1-2020*, pp. 1-20, 2021.

[21] S. Li, T. Zhang, B. Yu, and K. He, "A provably secure and practical PUF-based end-to-end mutual authentication and key exchange protocol for IoT," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 5487-5501, 2021.

[22] M. Saleh "Secure optimized request zone location-aided routing protocols with Wi-Fi direct for vehicular ad hoc networks," *Journal of Communications*, vol. 17, no. 3, pp. 156-166, 2021.

[23] M. A. Al-Shabi, "A survey on symmetric and asymmetric cryptography algorithms in information security," *Int. Journal of Scientific and* Research *Publications (IJSRP)*, vol. 9, no. 3, pp. 576-589, 2019.

[24] P. Morrison, D. Moye, R. Pandita, and L. Williams, "Mapping the field of software life cycle security metrics," *Information and Software Technology*, vol. 102, pp. 146-159, Oct. 2018.

[25] M. Mathankumar, S. Karthikeyani, S. G. Kumar, N. Mahesh, N. J. Savitha, and R. Rajaguru, "An efficient dynamic key generation architecture for distributed wireless networks," in *Proc. Third Int. Conf. on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, 2021, pp. 157-160.

[26] J. Jin and K. Kim, "3D CUBE algorithm for the key generation method: Applying deep neural network learning-based," *IEEE Access*, vol. 8, pp. 33689-33702, 2020.

[27] A. Sarkar, "A symmetric neural cryptographic key generation scheme for IoT security," *Applied Intelligence*, 2022.

[28] S. Pamarthi and R. Narmadha, "Adaptive key management-based cryptographic algorithm for privacy preservation in wireless mobile adhoc networks for IoT applications," *Wireless Personal Communications*, vol. 124, no. 1, pp. 349-376, 2022.

[29] A. Singh, A. and K. Jain, "An efficient secure key establishment method in cluster-based sensor network," *Telecommunication Systems*, vol. 79, no. 1, pp. 3-16, 2022.

[30] G. Mehmood, M. S. Khan, A. Waheed, M. Zareei, M. Fayaz, T. Sadad, and A. Azmi, "An efficient and secure session key management scheme in wireless sensor network," *Complexity*, 2021.

[31] G. Mehmood, M. Z. Khan, H. U. Rahman, and S. Abbas, "An efficient and secure session key establishment scheme for health-care applications in wireless body area networks," *Journal of Engineering and Applied Sciences*, vol. 37, no.1, pp. 9-18, 2018.

[32] S. Ali, A. Humaria, M. S. Ramzan, *et al*., "An efficient cryptographic technique using modified Diffie–Hellman in wireless sensor networks," *Int. Journal of Distributed Sensor Networks*, vol. 16, no. 6. 2020.

[33] A. Shafique and J. Shahid, "Novel image encryption cryptosystem based on binary bit planes extraction and multiple chaotic maps," *The European Physical Journal Plus*, vol. 133, no. 8, pp. 1-16, 2018.

[34] Y. Sheng, J. Li, X. Di, X. Li, and R. Xu, "An image encryption algorithm based on complex network scrambling and multi-directional diffusion," *Entropy*, vol. 24, no. 9, 2022.

[35] K. Suresh, P. Rajarshi, and S. R. Balasundaram, "Two-factor-based RSA key generation from fingerprint biometrics and password for secure communication," *Complex & Intelligent Systems*, vol. 8, pp. 3247-3261, Feb. 2022.

[36] R. Dwivedi, S. Dey, M. A. Sharma, and A. Goel, "A fingerprint based crypto-biometric system for secure communication," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 4, pp. 1495-1509, 2020.

[37] W. Cao, Y. Zhou, C. P. Chen, and L. Xia, "Medical image encryption using edge maps," *Signal Processing*, vol. 132, pp. 96-109, Mar. 2017.

[38] J. Qin, *et al*., "An encrypted image retrieval method based on Harris corner optimization and LSH in cloud computing," *IEEE Access*, vol. 7, pp. 24626-24633, 2019.

[39] X. Li, Y. Jiang, M. Chen, and F. Li, "Research on iris image encryption based on deep learning," *EURASIP Journal on Image and Video Processing*, vol. 126, 2018.

[40] J. Abukhait and M. E. Saleh, "A real-time multi-agent system for cryptographic key generation using imaging-based textural features," *Int. Journal of Information and Communication Technology*, vol. 14, no. 4, pp. 470-484, 2019.

[41] K. Sood, K. K. Karmakar, S. Yu, V. Varadharajan, S. R. Pokhrel and Y. Xiang, "Alleviating heterogeneity in SDN-IoT networks to maintain QoS and enhance security," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5964-5975, 2020.

[42] M. Saleh and L. Dong, "Real-time scheduling with security enhancement for packet switched networks," *IEEE Trans. on Network and Service Management*, vol. 10, no. 3, pp. 271-285, 2013

[43] M. Saleh and L. Dong, "Adaptive security-aware scheduling using multi-agent system," in *Proc. IEEE Int. Conf. on Communications (ICC)*, 2012, pp. 1149-1153.

[44] B. Mao, Y. Kawamoto, and N. Kato, "AI-based joint optimization of QoS and security for 6G energy harvesting internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7032-7042, 2020.

[45] R. Franzen. Kodak Lossless True Color Image Suite. Kodak. (1999). [Online]. Available: http://r0k.us/gaphics/kodak/.

[46] D. Guan, D. Xiang, X. Tang, L. Wang, and G. Kuang, "Covariance of textural features: A new feature descriptor for SAR image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 10, pp. 3932-3942, 2019.

[47] D. A. Clausi, "An analysis of co-occurrence texture statistics as a function of grey level quantization," *Can. J. Remote Sensing*, vol. 28, no.1, pp. 45-62, 2002.

[48] L. Soh and C. Tsatsoulis, "Texture analysis of SAR sea ice imagery using gray level co-occurrence matrices," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 37, no. 2, pp. 780-795, 1999.

[49] IEEE standard 754-2008. IEEE Standard for Floating-point Arithmetic, IEEE Computer Society, 2008.

Jafar Abukhait received his B.Sc. and M.Sc. degrees in Computer Engineering from Yarmouk University in 2005 and 2007, respectively. He received his Ph.D. degree in Electrical and Computer Engineering from Western Michigan University in 2012. In 2012, he joined the faculty of Tafila Technical University as an Assistant Professor of Electrical and Computer Engineering. He got promoted to Associate Professor in 2018. His research interests are mainly in digital image processing, feature extraction, pattern recognition, and classification Techni Networks, Real-Time Agent-Based Systems, and QoS for Heterogeneous Networks.

Ma’en Saleh (M’10) received his Ph.D. degree in Electrical and Computer Engineering from Western Michigan University in 2012. He joined the faculty of Tafila Technical University as an Assistant Professor of Electrical and Computer Engineering in 2012. He joined the ECE department at Baylor University, TX in 2016 as a postdoctoral researcher. He promoted to Associate Professor in 2018. His research interests include Real-Time Scheduling for Packet Switched Networks, Security in VANETs, Simulating Real-Time Networks, Real-Time Agent-Based Systems, and QoS for Heterogeneous Networks.