IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 7 July 2023, accepted 30 July 2023, date of publication 7 August 2023, date of current version 15 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3302529

RESEARCH ARTICLE

SAP: A Secure Low-Latency Protocol for Mitigating High Computation Overhead in WI-FI Networks

VINEETA JAIN[1], (Member, IEEE), ULF WETZKER[1], VIJAY LAXMI[2], MANOJ SINGH GAUR[2], (Member, IEEE), MOHAMED MOSBAH[3], AND DOMINIQUE MERY[4], (Member, IEEE)

[1]Division Engineering of Adaptive Systems (EAS), Fraunhofer Institute for Integrated Circuits IIS, 01187 Dresden, Germany
[2]Malaviya National Institute of Technology Jaipur, Jaipur 302017, India
[3]LaBRI, CNRS, Bordeaux INP, University of Bordeaux, 33405 Talence, France
[4]LORIA, University of Lorraine, 54518 Vandoeuvre-lès-Nancy, France

Corresponding author: Vineeta Jain (vineeta.jain@eas.iis.fraunhofer.de)

ABSTRACT The increase in popularity of wireless networks in industrial, embedded, medical and public sectors has made them an appealing attack surface for attackers who exploit the vulnerabilities in network protocols to launch attacks such as Evil Twin, Man-in-the-middle, sniffing, etc., which may result in economic and non-economic losses. To protect wireless networks against such attacks, IEEE 802.11 keep updating the protocol standards with new and more secure versions. There has always been a direct correlation between attacks and the improvement of protocol standards. As the sophistication of attacks increases, protocol standards tend to move towards higher security, resulting in a significant rise in both latency and computational overhead, and severe degradation in the performance of low-latency applications such as Industrial Internet of Things (IIoT), automotive, robotics, etc. In this paper, we make an attempt to highlight the importance of both latency and security in wireless networks from implementation and performance perspective. We make a review of existing IEEE 802.11 protocols in terms of security offered and overhead incurred to substantiate the fact that there is a need of a protocol which in addition to providing optimum security against attacks also maintains the latency and overhead. We also propose a secure and low-latency protocol known as Secure Authentication Protocol (SAP) which operates in two phases - registration and authentication, where the first phase is a one time process implemented using asymmetric cryptography and the second phase is implemented using symmetric cryptography. The protocol is structured in a way that it maintains the original structure of IEEE 802.11 protocols and performs both phases using fewer messages than existing protocols. By simulating the protocol using well-established OMNeT++ simulator, we proved that the proposed protocol incurs a low computation overhead, making it ideal for low-latency applications. We extensively verified the security properties of the proposed protocol using formal verification through widely-accepted Scyther tool. Finally, we perform a comparative analysis of SAP with existing IEEE 802.11 wireless network protocols to highlight the improvement.

INDEX TERMS Wireless network, security, low-latency, computation overhead, authentication, reauthentication.

I. INTRODUCTION

Nowadays, wireless networks have become one of the ubiquitous and fastest means of accessing the Internet across the globe. According to the Cisco Visual Networking Index for 2017 – 2022 [1], 51% of the global Internet Protocol (IP) traffic was predicted to be received from wireless networks by the end of 2022. This is mainly due to the ubiquity of wireless communication systems in the automotive, medical, military, IIoT and public sectors owing to the mobility and

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

flexibility offered by the wireless networks. In medical technology, wireless networked devices, such as electronic medical records, physiological monitoring devices (wearables) or diagnostic equipment, are already being used in large numbers. In vehicles, the use of wireless communication is not limited to infotainment systems. Vehicular communication supports the driver via computer-assisted safety warnings and traffic information from external sources and inside the vehicle sensor information is transmitted wirelessly for condition monitoring. With the digitalization of the automation industry, the number of wirelessly networked devices has increased significantly. Rigid structures such as conveyor belts and overhead cranes are increasingly being replaced by intelligent, automated guided vehicle (AGVs). Augmented reality (AR) supports the worker in carrying out individual work steps by providing work and safety instructions via wirelessly connected data glasses. With the increase in data volume-dependent costs and international roaming charges on 4*G* networks, the popularity of public free Wi-Fi networks has increased for the use of data, voice and video services, resulting in deployment of Access Points (APs) in public places such as airports, railway stations, cafes/restaurants, etc. As per the Cisco Annual Report for 2018-2023 [2], IIoT devices will witness a 2.4 fold growth from 6.1 billion in 2018 to 14.7 billion in 2023 [2]. Most of the above applications have very high security and availability requirements [3] that must be met by the communication system.

With the increasing popularity of wireless networks, it is also becoming a prime target of attackers, posing a dangerous threat to the safety of users. Some of the contemporary attacks include *eavesdropping* where an attacker actively or passively sniffs the information transmitted between clients and AP and then uses brute forcing and cryptanalysis techniques to decrypt the encrypted information, *Evil Twin (ET)* attack where an attacker deploys a rogue AP mimicking the genuine characteristics (such as SSID, BSSID, passphrases, etc.) of legitimate AP in the network to fool clients to connect to the ET allowing him to hijack sessions, intercept network traffic, push malicious payloads etc., *Man-in-the-middle (MITM)* attack where an attacker authenticates itself to both client and AP as AP and client, respectively by sniffing and redirecting authentication messages to maliciously locate itself between AP and client so that all the traffic can flow through the attacker which can lead to information loss, malware installation, financial loss, remote control, etc., and *replay* attack where an attacker maliciously uses the previously transmitted authentication messages to gain unauthorized access to the network. Launching these attacks in medical, military or industrial environments for sniffing or tampering the transmitted information could have serious consequences, including economic espionage, operational failure, physical damage, environmental harm, and injury or loss of life.

A. MOTIVATION

To protect wireless networks from such attacks, IEEE 802.11 has launched several protocols. Table 1 shows a summary of existing IEEE 802.11 standard protocols for authentication in wireless networks.

Wired Equivalent Privacy (WEP) is the first protocol launched by IEEE 802.11 in 1997, which uses password-based authentication where the password is a 40-bit static key already known to all the clients. Further, WEP applies Rivest Cipher 4 (RC4) stream cipher for encryption using 24-bit Initialization Vector (IV). Due to the unencrypted transmission of authentication messages, smaller key size and reuse of IVs [4], WEP is found vulnerable to MITM, ET and replay attacks [5].

To overcome the shortcomings of WEP, IEEE 802.11 launched **WI-FI Protected Access (WPA)** protocol in 2003. The aim was to fix the limitations of WEP without upgrading the hardware. WPA uses password-based authentication, where the password is a passphrase also known as pre-shared key (PSK). WPA applies Temporal Key Integrity Protocol (TKIP) for encryption which uses the RC4 algorithm and introduces the concept of 4-way handshake after the authentication and association phases. In the 4-way handshake, all types of keys used for encryption and transmission are generated using the PSK. Due to the use of RC4 for encryption and similar passphrase for all the clients, it is found vulnerable to offline dictionary attacks [6]. Once the attacker cracks the passphrase, launching MITM and ET attacks is a cakewalk.

Further in 2004, IEEE 802.11 introduced **WPA2** protocol which is an improved version of WPA protocol. WPA2 uses Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP) which utilizes Advanced Encryption Standard (AES) for encryption. In WPA2, both client and AP share a passphrase known as Pairwise Master Key (PMK), which is used to generate Pairwise Transient Key (PTK) for encrypting the user sessions. The usage of high-level encryption reduces the chance of cryptanalysis, but still the offline dictionary attack and ET attacks are possible [7].

In 2018, IEEE 802.11 has introduced **WPA3** protocol. For public networks, WPA3 uses opportunistic wireless encryption (OWE) mode also known as enhanced open WI-FI network [8]. In this mode, there is no pre-shared information between AP and client. Both the entities exchange their pubic keys to generate a shared secret key, i.e., PMK using the Elliptic Curve Diffie-Hellman (ECDH) algorithm. The derived PMK is then utilized in the four-way handshake mechanism to generate session keys, i.e., PTK. For personal networks, WPA3 provides an extra layer of security (in addition to WPA2) in the form of simultaneous-authentication-of-equals (SAE) handshake (a variant of the dragonfly handshake mechanism) and this standard is called as WPA3-personal. During SAE handshake, the passphrase shared between client and AP is converted into a high entropy key (PMK). Further, this key is used to produce PTK during four-way handshake mechanism. The computational overhead of WPA3-personal is relatively very high given the complexity of the SAE handshake [9].

TABLE 1. A summary of existing IEEE 802.11 standard protocols for authentication in wireless networks.

Protocol	Release date	Encryption	Advantages	Limitations
WEP	1997	RC4 stream cipher	Simple Implementation. Support for legacy devices.	No support for mutual authentication, making it vulnerable to MITM attacks. Weak security due to small key size. Limited number of available keys due to the use of 24-bit IV field.
WPA	2003	TKIP	Introduces four-way handshake for key generation and mutual authentication using pre-shared passphrases.	Vulnerable to offline dictionary attacks due to the use of RC4 and pre-shared passphrases. Once the passphrase is cracked, other network attacks like ET and MITM are easy to launch.
WPA2	2004	AES-CCMP	Use of AES encryption makes it more secure.	Still uses pre-shared passphrases for authentication and key generation, making it vulnerable to offline dictionary attacks, ET and MITM attacks.
WPA3-OWE	2018	ECDH and AES-CCMP	Does not use any pre-shared passphrases for authentication and key generation.	Vulnerable to ET attacks due to the use of Trust-on-first-use (TOFU) model for authentication.
WPA3-personal	2018	SAE handshake	Enhanced security for personal networks by using secure SAE handshake for key generation.	Computationally expensive due to the use of complex SAE handshake, making it unsuitable for low-latency applications.
802.1X	2001	EAP	Most secure protocol due to the use of unique certificates or credentials for every user for authentication.	Exchanges a large number of messages for authentication making it unsuitable for low-latency applications. Requires RADIUS server and certificate management, adding complexity and costs to the network setup.

802.1X protocol is an IEEE Standard for Port-Based Network Access Control (PNAC) which uses unique certificates or credentials for every user to authenticate eliminating the reliability on single password for authentication. In addition to client and AP, 802.1X also requires a RADIUS[1] server and identity provider for authentication. The RADIUS server verifies the identity of a client by communicating with the identity provider (a directory containing user credentials/certificates information). Although 802.1X is the most secure protocol, the number of messages exchanged for authentication are way too high for use in wireless networks. The typically fluctuating transmission conditions of a WiFi network occasionally lead to situations in which a new connection setup is required. Under such conditions, long connection setup procedures inevitably lead to greatly increased latency in the network.

WPA3 protocol also introduces **WPA3-enterprise** for high-security Wi-Fi networks such as in government, defense and finance. It includes an additional 192-bit security while still using 802.1X as the base protocol. Further, it adds an additional requirement of certificates for RADIUS servers along with clients. While this has added significantly to the security of the connection, it complicates and lengthens the authentication procedure, makes it computationally more expensive, and is thus not suitable for low-latency wireless networks.

To comprehensively protect Wi-Fi networks against attacks, vulnerabilities in the IEEE 802.11 specification were addressed by new and improved security mechanisms. Although the IEEE protocols provide security against attacks, they result in increased communication and computation overhead, making the authentication process more time-consuming. For a number of low-latency applications, such as heath-care, intelligent transportation system, robotics, AR, etc., this is unacceptable as it would lead to significantly increased downtime during operation leading to unavailability. Due to increasing concerns about the latency and the associated reliability of the IEEE 802.11 standard, many sectors have started to replace their wireless networks with private cellular networks [10]. This leads to an increase in their overall cost as private networks require greater upfront investment. Moreover, for constrained devices such as embedded and IIoT devices, the high computation overhead of security processes further increases these latency issues. Therefore, a lightweight protocol is needed that provides optimal protection against current network attacks (such as ET, MITM, replay and sniffing attacks) while keeping latency and computation overhead as low as possible.

B. CONTRIBUTION

In this paper, we propose a secure low-latency protocol named *Secure Authentication Protocol (SAP)* which provides security against contemporary network attacks through a secure authentication and reauthentication mechanism. We define *authentication* as the first attempt of the client to get authenticated to an AP. Any subsequent authentication attempts between a client and an AP are defined as *reauthentication*. SAP employs Elliptic Curve Cryptography

[1] It stands for Remote Authentication Dial-In User Service.

(ECC) for key distribution and authentication, and symmetric encryption for reauthentication and session establishment. Although the usage of ECC and symmetric encryption in security protocols is already established, the novelty lies in how and where they are deployed. SAP ensures that even with the use of cryptographic primitives, the protocol remains lightweight and consumes less number of messages yet provides optimum level of security required. SAP has the following advantages: (i) In the proposed protocol, key generation and distribution between client and AP is performed only once when they get associated for the first time. The process utilizes fewer messages than existing protocols and do not require any additional servers, pre-shared knowledge or large number of message exchanges making it suitable for low-latency applications and embedded systems, (ii) once the key distribution and mutual authentication occurs between client and AP, they cache the relevant connection information. For future sessions, the client and AP only undergo reauthentication using the cached information. This makes the process resource-saving, computationally efficient and fast, and (iii) the proposed protocol does not modify the original structure of 802.11 protocol stack. Thus, the deployment is easy on the user side. In short, the paper makes the following contributions:

- In this paper, we highlight the importance of low-latency for the wireless network protocols and propose a protocol known as *SAP*, which in addition to providing security against contemporary network attacks also keeps the overhead and delay maintained.
- We propose to use both symmetric and asymmetric cryptography in the protocol in a way that it preserves the original structure of IEEE 802.11 protocols, guarantees mutual authentication and secure key distribution, and exchanges less number of messages incurring low computation overhead in session establishment between client and AP.
- We intensively tested our protocol using formal verification to test the security properties and simulation for network performance parameters. We also compared our proposed protocol with the previous IEEE 802.11 standard protocols to highlight the improvement obtained via the proposed protocol.

The organization of the paper is as follows: Section II discusses the existing studies and works related to security-performance tradeoff in wireless networks to establish the motivation for a low-latency secure protocol. Section III discusses the various types of keys and network assumptions followed by the proposed protocol, and provides a detailed description of the protocol. Section IV theoretically analyzes the proposed protocol in various aspects such as security analysis and mutual authentication between client and AP. Section V evaluates the security aspects of the proposed protocol by formally verifying SAP using Scyther. Section VI explains the practical demonstration of SAP using OMNeT++ simulator. Section VII compares SAP with the existing standard protocols. Section VIII concludes the paper.

II. RELATED WORK

When Wi-FI was introduced in 1997, it included the WEP protocol as a security standard. This embarked use of cryptography in wireless networks. However, WEP could not survive for long and was broken in 2001 [11] (details are discussed in Section I-A). This led to the introduction of new protocol standards.[2] With time, as attackers started using advanced and concealed ways of attacking wireless networks, the security of Wi-Fi standards kept increasing.

A comparable pattern was observed in the domain of integrated circuit design. System-on-Chips (SoCs) use Network-on-Chips (NoCs) for fast and efficient communication between on-chip Intellectual Property (IP) cores [12]. SoCs suffer from attacks like malicious IPs, Hardware Trojans, etc. [13], which are capable of launching attacks such as eavesdropping, injection, etc. To prevent such attacks, the use of symmetric and asymmetric cryptography was introduced in NoCs [13]. However, in the hardware domain, a lot of factors such as chip area, propagation delays, energy consumption, etc., have to be considered when applying security methods.

In the wireless sector, on the contrary, the focus was predominantly on stronger security, which is why the computational complexity and latency increased continuously in the course of the evolution of the protocols. Jindal and Singh [14] performed a quantitative experimental analysis for different variants of WPA and WPA2 to observe their impact on throughput when implemented in wireless networks. The study revealed that an increase in response time and decrease in throughput is observed when protocols with heavy encryption algorithms are implemented in the network. A similar study performed in [15] compares the CPU utilization between the security methods implemented in WPA2 and WPA3 to conclude that the network performance degrades even further in WPA3 as it incorporates computationally expensive algorithms. This has captured the attention of research community who has started focusing on proposing cost effective security protocols.

Pandey et al. [16] modified the 802.1X protocol by replacing EAP-TLS mode, where every user is provided a unique digital certificate for authentication, with token based authentication using a set of pre-shared keys shared between client, AP and the server, thus reducing the key exchange computation overhead. Although the protocol is cost-effective, the protocol does not guarantee perfect forward secrecy [17]. Further, the use of pre-shared keys makes them vulnerable to offline attacks and cryptanalysis [18]. Similarly, the protocols proposed in [19], [20], [21], and [22] also modify 802.1X protocol for reducing the number of message exchanges to make it lightweight. Firstly, these protocols either use pre-shared keys/passwords or assume the medium of transmission while exchanging keys is secure or does not guarantee freshness of nonces, and secondly, they use 802.1X protocol as a base protocol which already exchanges a large number

[2] All protocol standards are discussed in detail in Section I-A.

of messages (details in Section VII). So, even if these protocols reduce a few message exchanges, the computation overhead incurred by these protocols is still high. Several works like [23] and [24] modify WPA2-PSK protocol for making it secure against offline dictionary attacks by introducing additional keys and encryption algorithms for making the protocol more secure. Since their intention was focused only towards making the protocol more secure, the computation overhead incurred is high making them not suitable for low latency applications. With a vast increase in use of low-power devices like IoTs and IIoTs, the need of a protocol taking into account the security-performance tradeoff is indispensable.

III. THE PROPOSED PROTOCOL

To address the issues present in the existing standard protocols, we propose **S**ecure **A**uthentication **P**rotocol (SAP). SAP neither uses open nor password-based authentication. SAP employs Elliptic Curve Cryptography (ECC) to generate and exchange keys, and symmetric encryption scheme to encrypt transmitted messages. ECC is chosen because it has outperformed the existing key generation algorithms such as RSA, owing to its shorter key size and small computational overhead [25]. Short key size makes ECC faster and suitable for small and embedded devices. Further, SAP uses Advanced Encryption Standard Counter Mode with Cipher Block Chaining Message Authentication Protocol (AES-CCMP) [4] for symmetric encryption of the messages, as AES-CCMP provides a high level of security for encryption, used by all standard protocols (such as WPA2 and WPA3) and not been proved vulnerable to attacks [4]. By incorporating these cryptographic and encryption schemes, SAP assures mutual authentication, encrypted communication, secrecy against eavesdroppers and resistance to attacks.

A. PRELIMINARIES

In this subsection, we present a concise description of Elliptic Curves and ECC and discuss the various types of keys and network assumptions followed by the proposed protocol.

1) ELLIPTIC CURVES

The elliptic curve over a finite field is defined by

$$y^2 = \{x^3 + ax + b\} \text{mod} \{p\} \tag{1}$$

It has domain parameters *(p, a, b, G, n, h)* where,

- p = prime number specifying the size of finite field,
- a, b = curve parameters,
- G = Generator Point (generates a cyclic subgroup),
- n = ord(G) (size of subgroup),
- h = cofactor = $\frac{|E(Z/pZ)|}{n}$ (ideally 1), where $E(Z/pZ)$ represents elliptic curve defined over Z (integers) modulo p.

Suppose an elliptic curve is defined over integer modulo p as $E(Z/pZ)$ and Q, $P \in E(Z/pZ)$, where P and Q are points on the curve such that

$$P = kQ = Q + Q \ldots k \text{ times} \tag{2}$$

According to **Elliptic Curve Discrete Logarithm Problem (ECDLP)**, the computation of P is simple when k and Q are known. However, given P and Q, the calculation of k is computationally challenging and expensive. This is the basis of ECC.

2) ECC

ECC encodes the message to a point on the curve specified by Eq. 1. The original message can be retrieved by decoding the point. For encoding and decoding the points, entities require key pairs. Suppose A and B have private keys as n_a and n_b respectively. The public keys of A and B are derived as:

$$P_a = n_a G \tag{3}$$

$$P_b = n_b G \tag{4}$$

When A wants to send a message m, he needs to perform two actions - (1) encode the message to a point T (ϕ_T) on the curve, and (2) create any random value u. The process of encoding a message m to a point ϕ_T is known as *mapping*, and the process of decoding ϕ_T to m is known as *reverse mapping* [26]. The mapping operations are performed by mapping function F such that

$$F(m) \rightarrow (x, y) \in E_p(a, b) \tag{5}$$

where m is the message and (x, y) are points on the curve $E_p(a, b)$ (as described in Eq. 1). Further, A encrypts the point ϕ_T using P_b and random variable u as:

$$Enc \rightarrow A : \{uG, \phi_T + uP_b\} \tag{6}$$

A sends this message to B. When B receives the message, he can decrypt the message by using uG and B's private key n_b.

$$Dec \rightarrow B : \phi_T + uP_b - n_b(uG) \tag{7}$$

Using Eq. 4, Eq. 7 can be rewritten as:

$$Dec \rightarrow B : \phi_T + u(n_b G) - n_b(uG) \tag{8}$$

By rearranging the term $u(n_b G)$, it can be rewritten as:

$$Dec \rightarrow B : \phi_T + n_b(uG) - n_b(uG) = \phi_T \tag{9}$$

B calculates $n_b(uG)$ to remove uP_b. Hence, no one other than B can decrypt ϕ_T. Further, B decodes ϕ_T using reverse mapping function to obtain the original message m. Thus, the attacker's attempt of obtaining the message m by eavesdropping the communication remains unsuccessful because they don't possess the private keys. The proposed protocol uses the same concept for encryption and decryption using ECC.

3) KEYS USED IN SAP

The keys play a significant role in ensuring the security of the proposed protocol. The following keys are used in authentication and reauthentication phases of SAP:

- *Public-Private Key Pair*: AP produces a public-private key pair using ECC, and the public key of AP is known to everyone in the network.

- *Encryption-Decryption Key Pair*: Client produces encryption-decryption key pair using ECC. The functionality of encryption-decryption key pair is similar to public-private key pair in a way that the information encrypted by encryption key can only be decrypted by decryption key. But the difference is that unlike the public key, encryption key of client is not public in the network.
- *Master Key (MK)*: MK is uniquely generated by AP for each client and exchanged only once between the client and AP during their first connection attempt. MK is cached by both the parties as MK is used as a reauthentication parameter for further connection attempts between the client and AP.
- *Session Keys*: They are freshly produced for each session between client and AP, and used for encrypting the communication between them.

4) ASSUMPTIONS

Following are the assumptions in proposed SAP protocol:

- The AP has a valid public key certificate[3] issued by a trusted and verified Certification Authority (CA).
- The client is loaded with a list of trusted CA certificates.
- The key-pairs[4] for network entities are generated only once.
- The client and AP have sufficient storage and mechanism for MK Caching.
- The AP and client possess encoder/decoder to transpose an elliptic curve point into information.
- The protocol is public.
- The AP, attacker and client are in the same network.

The attacker possesses the following characteristics:

- The attacker can conduct active as well as passive attacks.
- The attacker has access to the public key of the AP and ECC domain parameters.
- Any authentic client in the network can be a target of the attacker.

B. PROTOCOL OVERVIEW

SAP operates in two phases - *Registration* and *Authentication*. In the registration phase, client and AP generate their key pairs (explained in Section III-A3) and use them to securely exchange master key (MK). The MK is then further used to mutually authenticate and generate session keys (SK) in the authentication phase. The registration phase is a one-time process borne by AP and client during their first association. Once a client is registered with an AP, it undergoes authentication process to start the data transmission. Figure 1 shows an overview of the proposed protocol.

[3]Public key certificate, also known as a identity certificate or digital certificate, is an electronic document issued by a Certification Authority (CA) to prove the ownership of a public key. It contains name of the certificate holder, public key of the holder and the digital signature of a CA for authentication.

[4]Public-private key pair for AP and encryption-decryption key pair for clients.

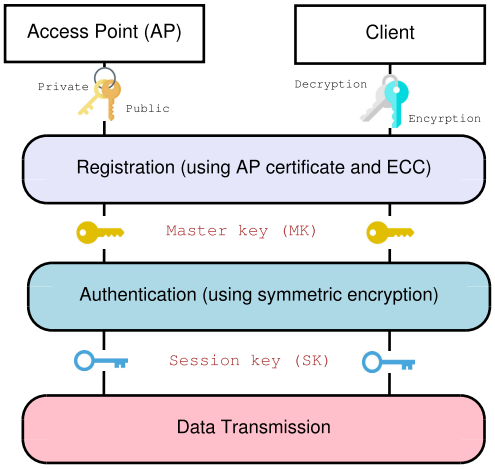

FIGURE 1. Overview of the proposed protocol.

TABLE 2. Notations with their descriptions used by SAP.

S.No.	Notation	Description
1.	C	Client
2.	AP	Access Point
3.	n_c	Decryption key of client
4.	P_c	Encryption key of client
5.	n_{AP}	Private key of AP
6.	P_{AP}	Public key of AP
7.	ϕ_T	Point T on the elliptic curve
8.	m	Message m
9.	m′	Decrypted message m
10.	P_{MK}	Master Key (MK)
11.	K_{sk}	Seed key
12.	K_{se}	Session key
13.	T	Timestamp

1) REGISTRATION PHASE

The registration phase is a one-time process which occurs when the client tries to connect to an AP for the first time. It consists of the following steps:

(R_1) **Beacon Frame**: The AP broadcasts beacon frames in the network embedded with its public key certificate issued by a legitimate and verified CA containing the ECC domain parameters and public key of the AP.

(R_2) **Probe Request:** The client, on receiving the beacon frame, verifies the certificate of the AP. It checks whether it implicitly trusts the certificate or it is trusted and verified by one of various CAs that it also implicitly trusts. If the client detects any problem in the certificate, i.e., either expired or hostname is different or not issued by any verified CA, it rejects the beacon and begin searching for new APs in the network. Else, it extracts the ECC domain parameters from the certificate and using them, the client chooses a decryption key n_c and

produces an encryption key P_c as:

$$P_c = n_c G \tag{10}$$

Further, the client sends a probe request to the AP consisting of P_c and current timestamp value T_c by encoding it to a point T (ϕ_T) and encrypts ϕ_T using P_{AP} (as explained in Section III-A2) extracted from the certificate, such that $\phi_T = P_c||T_c$. The timestamp is included to prevent replay attacks. The message also includes the hash of the message to maintain integrity.

$$C \rightarrow AP : m_0 = \{kG, \phi_T + kP_{AP}\}, h(m_0) \tag{11}$$

(R_3) **Probe Response:** The AP possess a public-private key pair as $\langle P_{AP}\text{-}n_{AP}\rangle$, where:

$$P_{AP} = n_{AP} G \tag{12}$$

On receiving the message, AP decrypts it using n_{AP}. Let the decrypted message be represented as m_0'. AP matches T_c with the current timestamp. If T_c is verified, then it computes the hash of m_0', and matches against $h(m_0)$. If $h(m_0') = h(m_0)$, then it selects a point on the curve ϕ_J. Using ϕ_J and P_c, it produces a master key (MK) as:

$$MK = \text{SHA-256}(\phi_J||P_c) \tag{13}$$

AP encrypts the MK using P_c and sends it to the client by encrypting it to the point S such that $\phi_S = P_{MK}||T_{MK}$, where T_{MK} represents the current timestamp.

$$AP \rightarrow C : m_1 = \{uG, \phi_S + uP_c\}, h(m_1) \tag{14}$$

This method of key exchange is known as Elliptic Curve Diffie-Hellman (ECDH) algorithm.

(R_4) On receiving the message, the client decrypts it using n_c. Initially, it verifies T_{MK} and $h(m_1)$ to detect the legitimacy of the message. Further, it computes Master Key Identifier (MKID) as:

$$MKID = trucate_{64}\{h(P_c||P_{AP})\} \tag{15}$$

The client caches MK and MKID. Similarly, the AP also calculates MKID and caches MK and MKID.[5] IEEE 802.11 implements "Pairwise Master Key (PMK) caching" for WPA where a client and AP can cache a PMK for a certain period and reuse it during the 4-way handshake occurring at the time of reassociation to bypass potentially expensive authentication. We have implemented the concept of caching to bypass the process of registration during reauthentication. For further connections, the client directly sends the authentication request encrypted with MK to the AP. Figure 2 shows the steps involved in registration phase.

[5]When physical access is possible (stolen devices), the information stored in the memory must be protected from hardware attacks such as side-channel analysis, fault injection, etc., which is beyond the scope of this work. However, they can be prevented by methods such as data encryption, compiler based countermeasures, use of hardware security modules, etc. [27].

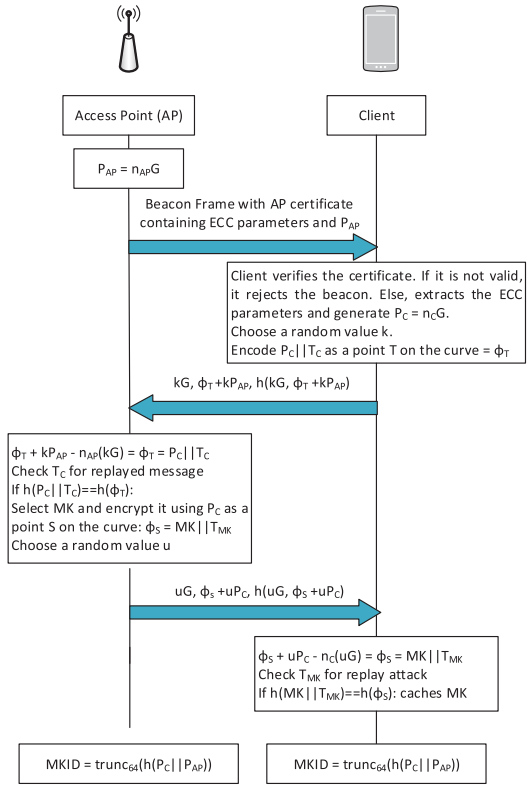

FIGURE 2. Registration phase.

2) AUTHENTICATION PHASE

This phase authenticates a client to the network. Whenever the client gets disconnected from the network, the reassociation begins with this phase, known as reauthentication. The session keys are produced during this phase, which are utilized for encrypting further communication. All the message exchanges in this phase are encrypted. The phase contains the following steps:

(A_1) **Auth Request**: The client generates a nonce value N_A and sends it to AP with the current timestamp value T_A by encrypting them with MK.

$$C \rightarrow AP : m_2 = (N_A, T_A)_{MK} \tag{16}$$

(A_2) **Auth Response**: On receiving Auth request, AP verifies T_A to check whether the message is genuine or any replayed message. Further, the AP generates a seed key K_{sk} and sends $\{N_A, K_{sk}, T_{sk}\}$ to the client encrypted with MK, where T_{sk} represents the timestamp value. The AP sends N_A again in the response to prove that the AP has successfully received the Auth request message and not any replayed message.

$$AP \rightarrow C : m_3 = \{N_A, K_{sk}, T_{sk}\}_{MK} \tag{17}$$

(A_3) Next, the client and AP produce session key using Password-Based Key Derivation Function 2 (PBKDF2) [28], which uses K_{sk} as the key and $N_A||MKID$ as the salt. It undergoes 4096 rounds of

encryption to produce a key of 128 bytes in length. We use PBKDF2 because the cryptanalysis attack is highly expensive for this function [28].

$$K_{se} = PBKDF2(HMAC - SHA256, K_{sk}, N_A||MKID, 4096, 128) \quad (18)$$

(A_4) **Auth Completion Request**: Client further generates a nonce value N_B, timestamp value T_{AB}, and sends $\{N_A, N_B, T_{AB}\}$ to the AP by encrypting it with new session key K_{se}.

$$C \rightarrow AP : m_4 = \{N_A, N_B, T_{AB}\}_{K_{se}} \quad (19)$$

(A_5) **Auth Completion Response**: The AP verifies T_{AB} and N_A, and acknowledges the correctness of the received message by sending N_B and T_B encrypted with K_{se}. This message proves that both the parties have correctly generated the session key. The purpose of nonces and timestamps in the entire communication is to determine the continuity of messages and prevent replayed messages.

Figure 3 shows the steps involved in authentication phase. Further, the client and AP proceed towards the association phase. Notably, all the subsequent transmitted messages are encrypted with the session key, which gets changed with every session[6] as during reauthentication new session keys are produced by the client and AP.

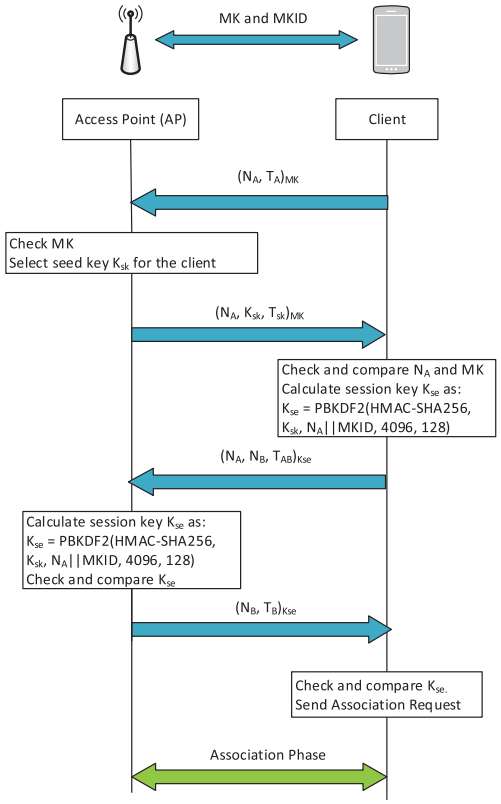

FIGURE 3. Authentication phase.

C. DISCUSSION

The proposed protocol uses a combination of asymmetric and symmetric cryptography in a way that it remains lightweight and yet provides optimum security against network attacks. In the registration phase of SAP, ECC is used to perform secure key exchange. As per the studies and experiments conducted in [29], [30], and [31], ECC is the most suitable asymmetric cryptographic algorithm for low-power devices such as IoTs, IIoTs, microcontrollers, etc., as it provides an equivalent level of security even with smaller key sizes, leading to less power consumption and faster computations. Therefore, it is applied in many secure communication protocols concerning low-power devices such as the ones proposed in [32], [33], [34], and [35]. These protocols use ECC for both registration and authentication phases, or in other words, for both key exchange and mutual authentication, which leads to higher computation overhead than the proposed protocol. The reason being, although ECC is faster than other asymmetric encryption algorithms such as RSA, it still requires more computing power than symmetric encryption algorithms [31], [36]. However, symmetric encryption algorithms alone are vulnerable to many network attacks, such as brute forcing, dictionary attacks, etc. Therefore, designing a protocol with the right balance of security and cost by ensuring the appropriate use of cryptographic algorithms is challenging. In the proposed protocol, we use ECC for the initial communication setup where keys are exchanged. Once both parties possess the master key, we use a symmetric encryption algorithm for further communication. This keeps the proposed protocol secure, fast and lightweight.

There also exist protocols such as WPA3-OWE which uses ECC during the initial handshake for generating the master key. However, the approach used in the proposed protocol is different. In WPA3-OWE, both the parties exchange public keys, and further ECDH algorithm is utilized for deriving PMK. The unencrypted exchange of public keys and not validating them before proceeding toward PMK generation make WPA3-OWE vulnerable to MITM and ET attacks. This proves that the mere adoption of ECC does not guarantee the resistance of a protocol against network attacks.

The registration phase of the proposed protocol is similar to HTTPS in a way that both AP and client verify the identity of other parties using certificates. But the difference is that for every session in HTTPS, the server send its certificate to the client for verification followed by the generation of session keys. Whereas in our protocol, the client verifies the certificate of AP only once, which reduces the computation time. Moreover, with the use of the proposed protocol, the encrypted communication will become a normal scenario which will enhance the security of the network.

As explained in Section II, the existing protocols tend to move towards heavy cryptographic algorithms to make them more secure, increasing the number of exchanged messages,

[6]A session here refers to a connection-delimited two-way link between the communication parties that allows information exchange.

resulting in high communication and computation overhead. For low-latency applications, this is unacceptable as it would lead to significantly increased downtime during operation, leading to unavailability. The significant advantage of the proposed protocol is that it performs registration and authentication phases not only by utilizing less number of messages than the existing protocols (details in Section VII) but also lightweight cryptographic algorithms that do not require any additional servers, pre-shared knowledge, or a large number of message exchanges, making it suitable for low-latency applications (for more details refer to Sections VI-B and VII). The only limitation of the proposed protocol is the use of certificates for AP authentication in the registration phase to ensure protection against MITM and ET attacks. However, this is done only once during the first association between client and AP. Moreover, the certificates are only possessed by AP and not by the clients, which makes it convenient for low-power and resource constrained clients such as IoT devices, microcontrollers, embedded devices, etc. So, even though the protocol introduces the use of certificates which adds additional cost to AP setup, low-latency, small computation overhead, and security from network attacks make it a reliable choice for sectors requiring security and availability as their key parameters.

IV. ANALYSIS OF SAP

This section theoretically analyzes SAP in various aspects such as mutual authentication and security analysis.

A. MUTUAL AUTHENTICATION BETWEEN CLIENT AND AP

For authenticating client in the network, the AP verifies the MK received by the client in Auth request message. If the MK matches with the one generated and transmitted by the AP to the client in the registration phase (encrypted with the encryption key of the client), the AP authenticates the client in the network.

Suppose an attacker A sends an Auth-request message to AP encrypted with MK':

$$A \rightarrow AP : (N'_A, T_A)_{MK'} \tag{20}$$

On receiving the message, the AP tries to decrypt the message with the MK cached for the respective client. Since the message is encrypted with MK' and not MK, the AP drops the message and does not send any Auth-response message further.

Similarly, the client also uses MK as a parameter for verifying the authenticity of the AP. The client sends Auth request message encrypted with the MK received from AP in the registration phase. If the AP can decrypt the message and send correct Auth response message, the client believes the legitimacy of the AP.

Suppose the attacker A captures the Auth request message sent by client to AP:

$$C \rightarrow AP : (N_A, T_A)_{MK} \tag{21}$$

Since the attacker does not possess MK, he is unable to decrypt the message. The attacker can try implementing the partial known-plaintext attack to crack MK as partial message (T_A) is already known to the attacker. However, due to the use of randomized nonces and AES-CCMP for encryption, the cryptanalysis is highly expensive [4]. Suppose, the attacker sends an Auth response message encrypted with MK':

$$A \rightarrow C : \{N'_A, K'_{sk}, T_{sk}\}_{MK'} \tag{22}$$

On receiving the message, the client tries to decrypt the message with the cached MK. When the client fails to decrypt, it drops the message and does not send any messages further.

Suppose, the MK shared between client and AP gets compromised. Consequently, the attacker successfully exchanges the Auth request and response messages with the client. But, when the attacker receives Auth completion request message from the client encrypted with the session key, he fails to decrypt the message because he is unable to produce correct session key. The reason being, the attacker does not have access to MKID generated in the registration phase. Thus, both client and AP can mutually authenticate each other in SAP and no third party can do this.

B. SECURITY ANALYSIS

In this subsection, we prove that the proposed protocol is able to prevent the exchanged messages from various types of network attacks, which we have formally proved in Section V. We assume that capturing private key of the AP and decryption key of the client is not possible by the attacker as nowhere in the communication they are being exchanged. Figure 4 shows the network model of SAP. In this model, three entities exist - client, AP and attacker. The figure shows the system setup of the network in the presence of the proposed protocol, the attacker's objectives and information possessed by the entities. In the figure, green-colored text represents publicly available information, black-colored text denotes the knowledge of the entities, and blue-colored text implies information being targeted by the attacker.

1) EAVESDROPPING

In SAP, all the exchanged messages are either encrypted using ECC or symmetric encryption to maintain end-to-end confidentiality of exchanged messages in the network. If the attacker eavesdrops the communication, he gets the encrypted frames which cannot be decrypted without the knowledge of private and decryption key (ECC), and MK and session keys (symmetric encryption). As already discussed, the attacker cannot obtain private and decryption keys because they are never transmitted. Thus, the attacker cannot generate MK and session keys as they are exchanged through messages encrypted with ECC keys. Hence, the transmitted messages in the network are secure from eavesdropping in the presence of the proposed protocol.

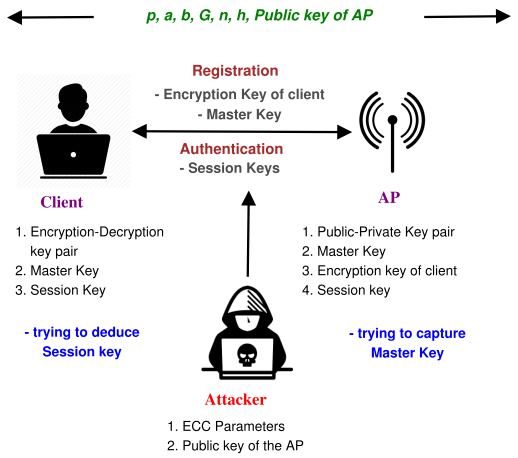

FIGURE 4. Network model showing network entities, their capabilities, attacker's objective, and system setup used in SAP.

2) REPLAY ATTACK

SAP is resistant to replay attack as it sends timestamp values ($T_c, T_{MK}, T_A, T_{sk}, T_{AB}, T_B$) with every message in the registration and authentication phase. Further, SAP also uses nonces in the authentication phase to prove the continuity of messages.

3) ET ATTACK

The objective of the attacker in ET attack is to force clients to get disconnected from the genuine AP and get connected to the ET so that the attacker can control the network traffic of the client. Suppose, in a network equipped with SAP; an ET disconnects a client from a genuine AP by sending deauthentication frames. The client tries to reauthenticate by sending Auth request frame. Although, the Auth request sent by the client is received by the ET, the ET is unable to read the contents of the message as it is encrypted with MK. Therefore, ET is unable to send correct Auth response message. An ET attack cannot be successful in the presence of SAP as an attacker needs private and public key of the AP and encryption key of the client to capture MK for successfully launching an ET attack.

4) MITM ATTACK

In this attack, the attacker tries to locate itself between client and AP, such that all the communication between them is through the attacker. Thus, the attacker can either intercept, replay or inject messages in the ongoing communication between the two parties. In our context, an attacker can perform MITM attack in two ways - (1) by launching ET attack and (2) by performing registration phase with the client. We have already proved that the ET attack cannot be launched in a network implementing SAP. Suppose, the client sends the probe request message to the AP. Attacker eavesdrops the communication and tries to forge the message to make the client connect to itself. However, the attacker cannot decrypt the probe request encrypted by the public key of AP as it does not have access to the private key of the AP. Nowhere in the exchanged messages, the private keys are shared. Moreover, we provide a public key certificate to the AP which makes the protocol resistant to MITM attack.

V. FORMAL VERIFICATION OF SAP

We use formal verification to automatically validate the security properties of the proposed protocol and illustrate that SAP thwarts network attacks. The formal verification of security protocols can be performed using model checking approach. It is based on evaluating the protocol by exploring all possible states and behaviors of the protocol. It runs multiple instances of the protocol simultaneously and analyzes whether the protocol satisfies security properties in all the instances or not.

To verify and analyze the security properties of the proposed protocol, we use Scyther. The reasons being manifold - (1) Scyther utilizes the unbounded model checking approach with confirmed termination which allows it to verify all possible states and behaviors of the protocol [37], (2) Scyther uses backward symbolic state search technique which empowers it to explore all type flaws and infinite state spaces [37] and (3) According to a study conducted in [38], Scyther is the fastest tool among the existing state-of-the-art tools. In case of an attack, it gives an attack scenario which provides a better understanding of the flaws in the protocol.

In the subsequent subsections, we explain the adversary model and security claims used in Scyther, and further, we discuss the modeling and verification of SAP using Scyther. Scyther uses *spdl* (Security Protocol Definition Language) format for describing the semantics of a protocol, which is explicitly invented for Scyther.

A. ADVERSARY MODEL

Scyther uses Dolev-Yao model as an adversary model which allows the adversary to replay, delete, breach, reroute, eavesdrop and process the content of the messages exchanged through the network. This model is predefined in the semantics of Scyther and thus, there is no need to define capabilities of an adversary for analyzing protocols in Scyther.

B. SECURITY CLAIMS

In Scyther, security properties are represented in the form of claims known as security claims. The adherence of a claim is checked by verifying whether the claim state is reachable or not during the protocol execution. Security claims in Scyther include:

- *Secrecy:* According to this claim, the messages exchanged over the network are not exposed to the attacker, even when the network is under full control of the attacker. The secrecy claim is expressed as $claim_L(R, secret, rt)$ which denotes, for the role R, rt should not be known to the adversary [37]. If rt is a session key, Scyther uses $claim_L(R, SKR, rt)$ to represent the secrecy of rt, where SKR stands for *Session Key Reveal*.

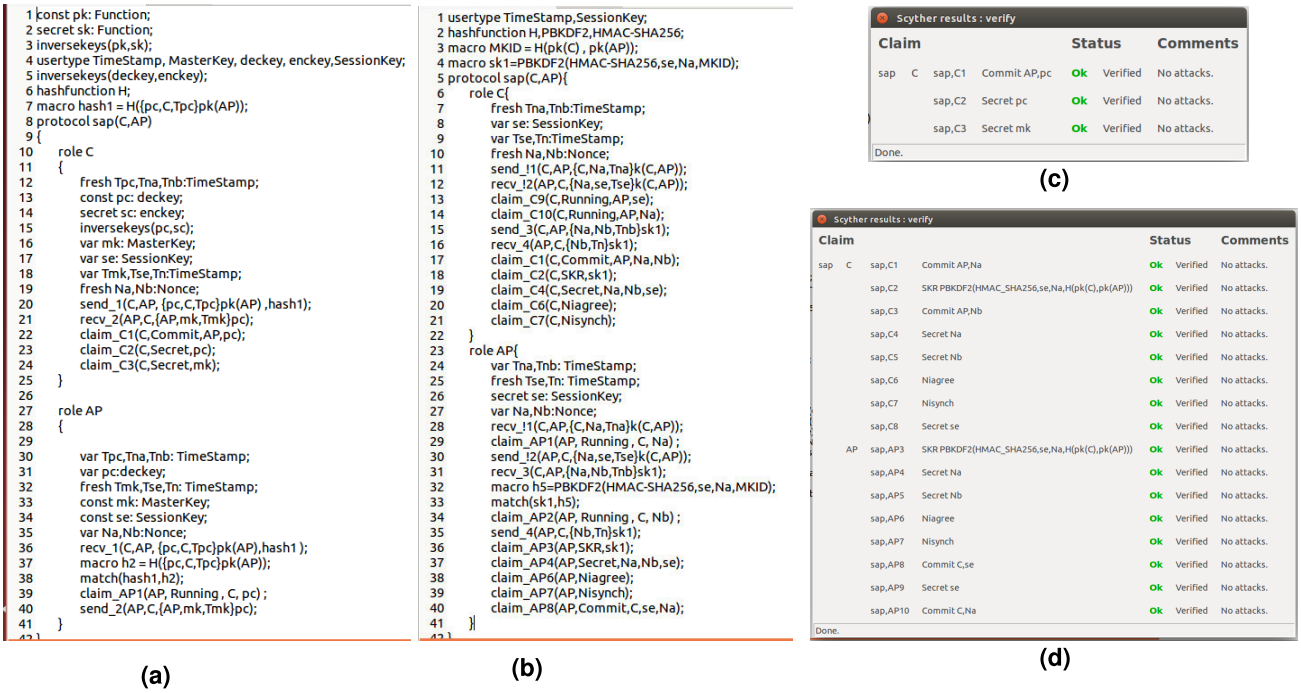

FIGURE 5. (a) spdl script for registration phase, (b) spdl script for reauthentication phase, (c) Scyther execution results for registration phase, and (d) Scyther execution results for reauthentication phase.

- *Mutual Authentication:* According to this claim, the communication must happen with the intended communication partner and not with the adversary. For verifying the authenticity of the communicating party, Scyther introduces the notion of *Synchronization*. This property states that the communication should occur between the expected, intended and genuine partners, and the protocol events should execute in the same way as described in the protocol specification. The *synchronization* claim is expressed as $Nisynch(R, Nisynch)$, where Nisynch stands for *Non-injective Synchronization*.
- *Agreement over exchanged messages*: Mutual authentication is not sufficient to judge whether the sent message is exactly same as the received message or not. It is also crucial to verify the integrity of the exchanged messages by checking the agreement of both the parties on the contents of exchanged messages. Scyther uses *commit* signal to verify the integrity of exchanged messages during protocol execution.

C. VERIFICATION OF THE PROPOSED PROTOCOL

The proposed protocol involves two parties - AP = *Access point* and C = *Client*. Table 3 represents the notations with their descriptions used in the verification of the proposed protocol using Scyther. The messages exchanged during the execution of the proposed protocol are represented according to the notations described in Table 3.

Registration

- $C \rightarrow AP : m_0 = \{pc,\ Tpc\}pk(AP),\ hash(m_0)$
- $AP \rightarrow C : m_1 = \{mk,\ Tmk\}pc,\ hash(m_1)$

TABLE 3. Notations with their descriptions used in the verification of the proposed protocol using Scyther.

S.No.	Notation	Description
1.	sc	Decryption key of client
2.	pc	Encryption key of client
3.	sk(AP)	Private key of AP
4.	pk(AP)	Public key of AP
5.	Na, Nb	Nonces
6.	se	Seed key
7.	sk1	Session key
8.	mk	Master key
9.	T	Timestamp

Authentication

- $C \rightarrow AP : m_2 = \{Na,\ Tna\}mk$
- $AP \rightarrow C : m_3 = \{Na,\ se,\ Tse\}mk$
- $C \rightarrow AP : m_4 = \{Na,\ Nb,\ Tnb\}sk1$
- $AP \rightarrow C : m_5 = \{Nb,\ Tn\}sk1$

We assume that revelation of $sk(AP)$ and sc to the attacker is not possible, as nowhere in the protocol specification, private and decryption keys are exchanged. Figure 5(a) and 5(b) represent the *spdl* scripts of registration and authentication phase of the proposed protocol, respectively. In Figure 5(b), the authentication phase represents the script used in reauthentication, where mk is replaced with $k(C, AP)$ as in reauthentication phase mk acts as a symmetric key shared between client and AP.

The following security *claims* are made in *spdl* scripts of registration and authentication phases of SAP:

- *claim*$(C, Secret, pc/mk)$: The encryption key of the client and master key generated by AP should not be revealed to the attacker for preventing MITM and ET attacks.
- *claim*$(C, Commit, AP, pc)$: Client C and AP should agree on the value of *pc* to proceed towards the authentication phase.
- *claim*$(AP/C, Secret, se/Na/Nb)$: The seed key and nonces used as an input to generate session key should not be leaked to the adversary to forbid attacker from generating the session key.
- *claim*$(AP/C, SKR, sk1)$: The generated session key $sk1$ should not be revealed to the attacker to avoid eavesdropping, MITM and message tampering attacks.
- *claim*$(C/AP, Commit, AP/C, Na/Nb/se)$: The AP and client should agree on the values of nonces *Na* and *Nb*, and seed key *se*, to ensure prevention from tampering attack.
- *claim*$(C/AP, Nisynch)$: For both the roles, the claim of synchronization should hold to ensure prevention from replay, ET and MITM attacks.

Figure 5(c) and 5(d) represent the Scyther execution results of the proposed protocol, which shows that no attack has been found in the protocol. Hence, the proposed protocol is secure from the network attacks.

VI. PRACTICAL PERSPECTIVE: SIMULATION OF PROPOSED PROTOCOL

To validate the appropriateness of the proposed protocol for low-latency applications by proving that it incurs a low computation overhead, we simulated SAP using the INET Framework extension of broadly accepted OMNeT++ simulator [39] on Windows 10 platform. The INET framework of OMNeT++ represents network devices such as hosts, switches, routers, APs, etc., as modules written in C++. It also contains devices configured with IEEE 802.11 network interfaces and represents several layers of the IP suite such as UDP, TCP, ARP and IPv4 protocols. We embed OpenSSL APIs (Application Program Interfaces) in INET framework to model the cryptographic operations (ECC, AES-CCMP and PBKDF2) of the proposed protocol.

A. SIMULATION SETUP

The setup for simulation consists of a network containing an AP modeled using *AccessPoint* compound module of INET, client modeled using *WirelessHost* compound module of INET, *configurator* module to assign IP address to the network entities, *radioMedium* module to send and receive packets for wireless nodes, *visualizer* module to visualize the packet transmission in the network, and *pcapRecorder* module to record the packets and further analyze them using packet analyzer tools such as Wireshark. Figure 6 shows the network setup used for simulation.

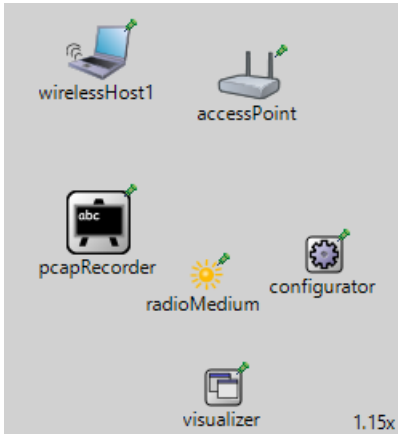

FIGURE 6. Simulation setup for the proposed protocol.

TABLE 4. Running time for primitive cryptographic operations.

Algorithm.	Action	Approximate Time Taken (in μs)
ECC	Encryption	2000
	Decryption	3000
	Key generation	1200
AES-CCMP	Encryption	400
	Decryption	500
PBKDF2	Key Generation	100

We modified the management layer frames of *AccessPoint* and *WirelessHost* modules to include the ECC encryption and decryption operations in beacon, probe request and probe response frames; AES encryption and decryption functions in authentication frames; and PBKDF2 function for session key generation. The UDP protocol is used for exchanging messages between the modules.

B. SIMULATION RESULTS AND DISCUSSION

The INET framework of OMNeT++ already contains a default implementation of the scanning and authentication process performed in IEEE 802.11 open networks. We modified the packet contents of the default implementation of IEEE 802.11 network protocol to implement the proposed protocol. A comparative analysis based on computation overhead (in *ms*) is performed between the proposed protocol and existing standard protocols to highlight the improvement of SAP in terms of latency. Further, an analysis of SAP based on packet size (in *bytes*) is also performed.

1) COMPUTATION OVERHEAD ANALYSIS

Since the proposed protocol performs key generation and encryption-decryption functions in registration and authentication phases, it incurs an additional computation time. In the simulation of the proposed protocol, the approximate computation time spent on cryptographic operations by using various cryptographic algorithms is shown in Table 4. For ECC operations, we use the standard NIST curve *Secp384r1*

TABLE 5. Comparison of SAP with existing protocols based on computation overhead incurred in Authentication process.

Protocol	Approximate Computation Overhead (in ms)
Open	NIL
WPA	$2 \times \eta + 4 \times \{\delta + \epsilon\} = 3.8$
WPA2	$2 \times \eta + 4 \times \{\delta + \epsilon\} = 3.8$
WPA3-OWE	$2\times\alpha+2\times\{\beta+\gamma\}+2\times\eta+4\times\{\delta+\epsilon\} = 16.2$
WPA3-personal	$40 \times \{\beta + \gamma\} + 2 \times \eta + 4 \times \{\delta + \epsilon\} = 203.8$
SAP	$2 \times \eta + 4 \times \{\delta + \epsilon\} = 3.8$

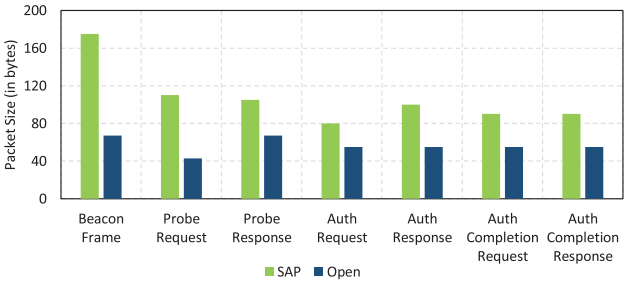

FIGURE 7. Packet size.

owing to the reason that for a highly secure system, a minimum of 384-bit key size is required [40].

Let the time taken to generate a key, encrypt and decrypt a message using ECC be represented as α, β and γ, respectively; time taken to encrypt and decrypt a message using AES-CCMP be denoted as δ and ϵ, respectively; and time taken to generate a key using PBKDF2 be represented as η.

During the registration phase in SAP, the client generates keys using ECC and all the message transmissions are encrypted and decrypted through ECC. Let the computation time spent during the registration phase be denoted as T_{reg}. It is calculated as:

$$T_{reg} = \alpha + 2 \times \{\beta + \gamma\} \tag{23}$$

Using the values in Table 4, T_{reg} is evaluated as approximately 11.2ms. During the authentication phase, SAP encrypts and decrypts the messages using AES-CCMP and produces session key using PBKDF2 algorithm. Let the computation time consumed during the authentication phase be represented as T_{auth}. It is computed as:

$$T_{auth} = 2 \times \eta + 4 \times \{\delta + \epsilon\} \tag{24}$$

The approximate value of T_{auth} is obtained as 3.8ms. Thus, the total overhead incurred while connecting to SAP is $15ms$ $(T_{reg} + T_{auth})$. However, the registration phase is a one-time process, so the overhead for reauthentication process of the proposed protocol is 3.8ms.

To compare the proposed protocol with existing standard protocols, we theoretically calculate the computation overhead incurred by them using the values depicted in Table 4. Table 5 represents a comparative analysis based on the computation overhead incurred in the authentication process. Since open networks do not implement any form of encryption or authentication, the computation overhead is NIL. Although the computation overhead of WPA and WPA2 are similar to the proposed protocol, they are vulnerable to offline dictionary and ET attacks (details are explained in Section I-A). In WPA3-OWE, ECC is used for PMK exchange and further, four-way handshake is performed to generate session keys. In contrast to our protocol which only performs MK exchange using ECC during the first association between client and AP, WPA3-OWE performs this in every association between both the parties leading to increase in computation overhead. In WPA3-PSK, client and AP undergo two rounds of four-way handshake - one handshake to convert shared password to a high entropy key (PMK) using hash-to-curve method of ECC and second handshake to generate session keys [9]. In hash-to-curve method, the output of the hash function is assumed to be the x-coordinate of a point on the EC curve and then using Eq. 1, the corresponding y-coordinate is searched via try-and-increment method up to maximum of 40 iterations [9]. This makes WPA3-PSK computationally very expensive.

For many applications, low-latency is of utmost importance, such as factory automation applications require latency between $0.25 - 10$ms, Intelligent Transport System (ITS) applications between $10 - 100$ms, healthcare applications between $1 - 10$ms and education applications require less than 10ms [41]. The proposed protocol not only incurs a low computational overhead but also is resistant to attacks, which makes it a suitable choice for such applications.

2) PACKET SIZE

Since the packets transmitted during registration and authentication phases in SAP implementation carries additional encrypted information for safely exchanging master and session keys, a comparison between the packet sizes of various frames under SAP and default open network implementation is required. Figure 7 shows the packet size comparison. The difference in packet sizes of frames under SAP and default open network INET implementation during registration and authentication phases is approximately 71B and 35B. We agree that the difference is not negligible, but if the trade-off between security offered by SAP in open networks and the packet sizes is considered, the difference can be ignored.

VII. COMPARISON OF THE PROPOSED PROTOCOL WITH EXISTING STANDARD PROTOCOLS

We compared SAP with the most widely used protocols WEP, WPA, WPA2, WPA3, and 802.1x based on the following characteristics in Table 6 - **Attack methods** successfully used against the protocols, **number of parties** involved in the connection establishment and the **total number of messages** exchanged during the process. According to Table 6, the open authentication protocol is vulnerable to all the attacks as it does not incorporate any type of authentication and encryption. WPA is susceptible to ET, MITM and message

TABLE 6. Comparative analysis of SAP with existing standard protocols.

Protocol	Eavesdropping	Replay attack	ET attack	MITM attack	Message Tampering attack	No. of parties involved	Total messages exchanged
Open	✗	✗	✗	✗	✗	2	7
SAP	✓	✓	✓	✓	✓	2	9
WPA	✓	✓	✗	✗	✗	2	11
WPA2	✓	✓	✗	✗	✓	2	11
WPA3-OWE	✓	✓	✗	✓	✗	2	13
WPA3-personal	✓	✓	✗	✓	✓	2	15
Credential-based 802.1X	✗	✓	†	✓	✓	3	22
Certificate-based 802.1X	✓	✓	†	✓	✓	3	15

✗= The protocol is vulnerable to the attack, ✓= The protocol is resistant to the attack, †= The protocol may or may not be vulnerable to the attack

tampering attacks due to the use of weaker encryption standards such as RC4. WPA2 is vulnerable to ET and MITM attacks due to the unencrypted transfer of nonces making it a prey to offline dictionary attack. WPA3-OWE is considered as the most secure protocol for public networks at the time of writing this paper. It is found secure against all attacks except message tampering and ET attacks because of unencrypted transmission of authentication messages and the use Trust-on-first-use (TOFU) model for authentication. The computational overhead of WPA3-personal is very high given the complexity of the SAE handshake [9]. Moreover, it undergoes two rounds of four-way handshakes before authentication leading to a high connection establishment time and thus, a high latency protocol. The credential-based 802.1X protocol is vulnerable to sniffing attack due to unencrypted over-the-air transfer of credentials. Most of the organizations including small-scale industries and tertiary educational institutes (TEIs) still use credential-based 802.1X protocol without enforcing the optional RADIUS server identity verification making them vulnerable to ET and sniffing attacks. As per the study conducted in [42], out of 7045 TEIs across 56 contries, 86% are vulnerable to credential theft and ET attacks. To overcome this problem, 802.1X introduced certificate-based protocol which uses unique client certificates for authentication. RADIUS server authentication by clients is not implemented in many networks due to the increased overhead caused by the certificate management, which makes it vulnerable to ET attacks. In addition, 802.1X exchanges a higher number of messages for authentication compared to SAP. The costs of implementing and operating an 802.1X based network are significant, as additional servers are required and the complexity of administration and certificate management is high.

From the Table 6, it can be seen that SAP exchanges the least number of messages compared to the presented protocols (except for open), which for directly results in reduced authentication and re-authentication time. In addition to this property, which is especially required in the area of low-latency applications, SAP offers very good protection against all evaluated threats. It assures end-to-end encrypted message transmission without any high-level expertise requirement for deployment. Although the protocol introduces the use of certificates for AP authentication which adds additional cost to AP setup, low-latency, small computation overhead and security from network attacks makes it a reliable choice for sectors requiring both security and availability as their key parameters. The use of ECC makes SAP suitable for embedded and IIoT devices as ECC incurs low computational overhead owing to its small key size.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we tried to emphasize the prominence of low-latency along with security in wireless networks for delay sensitive applications and studied the existing protocols in terms of computation overhead with respect to number of packets required for the authentication and reauthentication, and in terms of security with respect to traditional network attacks. We proposed a protocol comprising of two phases, where in the first phase cryptographic credentials are generated and securely exchanged and in the second phase, mutual authentication occurs. With simulation experiments, it is shown that the proposed protocol incurs low computation overhead and a lower authentication time. The protocol utilizes lightweight cryptographic primitives making it suitable for embedded and low-power devices. By using formal verification, we have justified resistance of SAP against network attacks and by performing comparative analysis, we demonstrate its advantages over the existing protocols. In future, the authors intend to imbibe trust management in the protocol with a strong focus on the deployment of certificates on embedded and IIoT devices so that client side attacks can be contained without additional expense. Further, the authors aim to reduce the computation overhead of the proposed protocol even more by exploring the recent lightweight symmetric encryption algorithms such as PRESENT.

Although 5G networks are already available and will be a performant alternative for many delay-sensitive applications, the initial investment and implementation costs for 5G networks are too high in many cases. They are not economically feasible for many small and start-up businesses as well as in

the hobbyist/consumer sector. In addition, the 5G standard is still under development and does not yet include all of the targeted features, so this technology is no replacement for widely used Wi-Fi based embedded platforms. In addition to the initial and ongoing operating costs, the flexibility of a communication standard will also play an important role regarding the acceptance. The use of the license-free *industrial, scientific, and medical (ISM)* frequency bands provides a high degree of freedom during development, which has led to products that are strongly tailored towards one specific application. It is likely that a large number of different wireless communication standards will continue to co-exist in the future, especially in certain specialized areas. For this reason, wireless network protocols need to be developed that place their emphasis on both security as well as low-latency and computational overhead. With the increasing use of IIoT and embedded devices, we would like to raise awareness among the research community about the importance of the interaction between security, latency, and computational overhead in the context of wireless network applications. The development of dedicated protocols would be an important step towards enhancing the use of wireless networks in industrial applications and would enable a variety of new applications.

REFERENCES

[1] *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2017–2022*. Accessed: Jun. 3, 2022. [Online]. Available: http://media.mediapost.com/uploads/CiscoForecast.pdf

[2] *Cisco Annual Internet Report (2018–2023) White Paper*. Accessed: Nov. 25, 2022. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[3] A. Frotzscher, U. Wetzker, M. Bauer, M. Rentschler, M. Beyer, S. Elspass, and H. Klessig, "Requirements and current solutions of wireless communication in industrial automation," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, Jun. 2014, pp. 67–72.

[4] C. Kohlios and T. Hayajneh, "A comprehensive attack flow model and security analysis for Wi-Fi and WPA3," *Electronics*, vol. 7, no. 11, p. 284, Oct. 2018.

[5] J. S. Park and D. Dicoi, "WLAN security: Current and future," *IEEE Internet Comput.*, vol. 7, no. 5, pp. 60–65, Sep. 2003.

[6] M. Waliullah and D. Gan, "Wireless LAN security threats & vulnerabilities," *Int. J. Adv. Comput. Sci. Appl.*, vol. 5, no. 1, pp. 1–12, 2014.

[7] O. Nakhila, A. Attiah, Y. Jin, and C. Zou, "Parallel active dictionary attack on WPA2-PSK Wi-Fi networks," in *Proc. IEEE Mil. Commun. Conf.*, Oct. 2015, pp. 665–670.

[8] S. Kwon and H.-K. Choi, "Evolution of Wi-Fi protected access: Security challenges," *IEEE Consum. Electron. Mag.*, vol. 10, no. 1, pp. 74–81, Jan. 2021.

[9] M. Vanhoef and E. Ronen, "DragonBlood: Analyzing the dragonfly handshake of WPA3 and EAP-PWD," in *Proc. IEEE Sym. Secur. Privacy (SP)*, Jun. 2020, pp. 517–533.

[10] *Industrial Wireless: Choosing Between Wi-Fi & cellular*. Accessed: Oct. 14, 2021. [Online]. Available: https://www.automate.org/industry-insights/industrial-wireless-part-3-choosing-between-wi-fi-and-cellular

[11] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the key scheduling algorithm of RC4," in *Proc. 8th Annu. Int. Workshop Selected Areas Cryptogr.*, Cham, Switzerland: Springer, 2001, pp. 1–24.

[12] S. Charles and P. Mishra, "A survey of network-on-chip security attacks and countermeasures," *ACM Comput. Surv.*, vol. 54, no. 5, pp. 1–36, Jun. 2022.

[13] A. Sarihi, A. Patooghy, A. Khalid, M. Hasanzadeh, M. Said, and A. A. Badawy, "A survey on the security of wired, wireless, and 3D network-on-chips," *IEEE Access*, vol. 9, pp. 107625–107656, 2021.

[14] P. Jindal and B. Singh, "Security-performance tradeoffs in a class of wireless network scenarios," *J. Netw. Syst. Manage.*, vol. 25, no. 1, pp. 83–121, Jan. 2017.

[15] D. Cahyadi, I. F. Astuti, and N. Nazaruddin, "Comparison of throughput and cpu usage between WPA3 and WPA2 security methods on wireless networks 802.11 n," in *Proc. AIP Conf.*, vol. 2482, 2023, pp. 1–13.

[16] A. Pandey, P. K. Pant, and R. C. Tripathi, "A system and method for authentication in wireless local area networks (WLANs)," *Proc. Nat. Acad. Sci., India Sect. A, Phys. Sci.*, vol. 86, no. 2, pp. 149–156, Jun. 2016.

[17] A. K. Yadav, M. Misra, P. K. Pandey, K. Kaur, S. Garg, and M. Liyanage, "LEMAP: A lightweight EAP based mutual authentication protocol for IEEE 802.11 WLAN," in *Proc. IEEE Int. Conf. Commun.*, May 2022, pp. 692–697.

[18] D. Felsch, M. Grothe, J. Schwenk, A. Czubak, and M. Szymanek, "The dangers of key reuse: Practical attacks on IPsec KE," in *Proc. 27th USENIX Secur. Symp.*, 2018, pp. 567–583.

[19] A. K. Yadav, M. Misra, P. K. Pandey, and M. Liyanage, "An EAP-based mutual authentication protocol for WLAN-connected IoT devices," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1343–1355, Feb. 2023.

[20] A. Kumar and H. Om, "A secure, efficient and lightweight user authentication scheme for wireless LAN," in *Proc. Int. Conf. Emerg. Trends Eng., Technol. Sci. (ICETETS)*, Feb. 2016, pp. 1–9.

[21] A. K. Yadav, M. Misra, M. Liyanage, and G. Varshney, "Secure and user efficient EAP-based authentication protocol for IEEE 802.11 wireless LANs," in *Proc. IEEE 17th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Dec. 2020, pp. 576–584.

[22] P. Kumar and D. Kumar, "A secure n-secret based client authentication protocol for 802.11 WLANs," *Telecommun. Syst.*, vol. 75, no. 3, pp. 259–271, Nov. 2020.

[23] K. S. Arikumar, A. D. Kumar, S. B. Prathiba, K. Tamilarasi, R. S. Moorthy, and M. M. Iqbal, "Enhancing the security of WPA2/PSK authentication protocol in Wi-Fi networks," *Proc. Comput. Sci.*, vol. 215, pp. 413–421, Jan. 2022.

[24] J. Noh, J. Kim, and S. Cho, "Secure authentication and four-way handshake scheme for protected individual communication in public Wi-Fi networks," *IEEE Access*, vol. 6, pp. 16539–16548, 2018.

[25] N. Mehibel and M. Hamadouche, "A new approach of elliptic curve Diffie–Hellman key exchange," in *Proc. 5th Int. Conf. Electr. Eng. - Boumerdes (ICEE-B)*, Oct. 2017, pp. 1–6.

[26] A. Sengupta and U. K. Ray, "Message mapping and reverse mapping in elliptic curve cryptosystem," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 5363–5375, Dec. 2016.

[27] *A Study on Hardware Attacks Against Microcontrollers*, German Federal Office Inf. Secur. (BSI), Germany, 2023.

[28] *PBKDF2*. Accessed: Mar. 15, 2018. [Online]. Available: https://en.wikipedia.org/wiki/PBKDF2

[29] M. Suárez-Albela, P. Fraga-Lamas, and T. Fernández-Caramés, "A practical evaluation on RSA and ECC-based cipher suites for IoT high-security energy-efficient fog and mist computing devices," *Sensors*, vol. 18, no. 11, p. 3868, Nov. 2018.

[30] S. M. Rajesh and R. Prabha, "Lightweight cryptographic approach to address the security issues in intelligent applications: A survey," in *Proc. Int. Conf. Intell. Data Commun. Technol. Internet Things (IDCIoT)*, Jan. 2023, pp. 122–128.

[31] M. Hutter, M. Feldhofer, and J. Wolkerstorfer, "A cryptographic processor for low-resource devices: Canning ECDSA and AES like sardines," in *Proc. 5th IFIP WG*. Cham, Switzerland: Springer, 2011, pp. 144–159.

[32] A. A. Alamr, F. Kausar, J. Kim, and C. Seo, "A secure ECC-based RFID mutual authentication protocol for Internet of Things," *J. Supercomput.*, vol. 74, no. 9, pp. 4281–4294, Sep. 2018.

[33] S. C. Rajkumar and K. R. Karthick, "Secure session key pairing and a lightweight key authentication scheme for liable drone services," *Cyber Secur. Appl.*, vol. 1, Dec. 2023, Art. no. 100012.

[34] K. Sowjanya, M. Dasgupta, and S. Ray, "An elliptic curve cryptography based enhanced anonymous authentication protocol for wearable health monitoring systems," *Int. J. Inf. Secur.*, vol. 19, no. 1, pp. 129–146, Feb. 2020.

[35] L. D. Tsobdjou, S. Pierre, and A. Quintero, "A new mutual authentication and key agreement protocol for mobile client—Server environment," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 2, pp. 1275–1286, Jun. 2021.

[36] B. Pearson, L. Luo, Y. Zhang, R. Dey, Z. Ling, M. Bassiouni, and X. Fu, "On misconception of hardware and cost in IoT security and privacy," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.

[37] C. J. F. Cremers, "Scyther: Semantics and verification of security protocols," Ph.D. dissertation, Inst. Program. Res. Algorithmics, Eindhoven Univ. Technol., Eindhoven, The Netherlands, 2006.
[38] C. Cremers and P. Lafourcade, "Comparing state spaces in automatic security protocol verification," in *Proc. 7th Int. Workshop Automat. Verification Crit. Syst.*, vol. 558. Amsterdam, The Netherlands: Elsevier, 2007, pp. 1–13.
[39] *OMNeT++ Discrete Event Simulator*. Accessed: Mar. 14, 2020. [Online]. Available: https://omnetpp.org/
[40] *Elliptic Curve Cryptography*. Accessed: Apr. 15, 2018. [Online]. Available: https://www.linuxjournal.com/content/elliptic-curve-cryptography
[41] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, "A survey on low latency towards 5G: RAN, core network and caching solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3098–3130, 4th Quart., 2018.
[42] M. H. Hue, J. Debnath, K. M. Leung, L. Li, M. Minaei, M. H. Mazhar, K. Xian, E. Hoque, O. Chowdhury, and S. Y. Chau, "All your credentials are belong to us: On insecure WPA2-enterprise configurations," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2021, pp. 1100–1117.

VINEETA JAIN (Member, IEEE) received the Ph.D. degree in computer science from the Malaviya National Institute of Technology at Jaipur, Jaipur, India. She is currently pursuing the Ph.D. degree with the Division Engineering of Adaptive Systems (EAS), Fraunhofer Institute of Integrated Circuits IIS, Germany. Her research interests include security and privacy with a special emphasis on wireless networks and mobile security.

ULF WETZKER received the Diploma degree in computer science and systems engineering from the Ilmenau University of Technology, Ilmenau, Germany. He is currently a Research Scientist with the Division Engineering of Adaptive Systems (EAS), Fraunhofer Institute for Integrated Circuits IIS, Germany. His research interests include wireless communication systems, data analytics, and machine learning, with a special focus on anomaly detection in wireless networks.

VIJAY LAXMI received the Ph.D. degree from the University of Southampton, U.K. She is currently a Faculty Member with the Department of Computer Science and Engineering, Malaviya National Institute of Technology at Jaipur, Jaipur, India. Her research interests include information security, Malware analysis, security, and QoS provisioning in wireless networks.

MANOJ SINGH GAUR (Member, IEEE) received the Ph.D. degree from the University of Southampton, U.K. He has been a Faculty Member with the Department of Computer Science and Engineering, Malaviya National Institute of Technology at Jaipur, Jaipur, India, and the Director of the Indian Institute of Technology, Jammu, India. His research interests include networks-on-chip, computer and network security, and wireless networks.

MOHAMED MOSBAH received the Ph.D. degree from the University of Bordeaux, in 1993. He is currently a Full Professor in computer science with the Polytechnic Institute of Bordeaux, France. He carries his research at the LaBRI, a research laboratory in computer science common with the University of Bordeaux and CNRS, where he is the Deputy Director. His research interests include distributed algorithms and systems, formal models, security, and ad hoc and sensor networks.

DOMINIQUE MERY (Member, IEEE) has been a Full Professor of computing science with the University of Lorraine, since September 1993. His research interests include proof-oriented system development for computer-based systems with higher levels of reliability and correctness, proof-based development of distributed algorithms, and the proof-based modeling of medical devices.