Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.052405

ARTICLE

# Design of an Efficient and Provable Secure Key Exchange Protocol for HTTP Cookies

**Waseem Akram[1], Khalid Mahmood[2], Hafiz Burhan ul Haq[3], Muhammad Asif[3], Shehzad Ashraf Chaudhry[4,5] and Taeshik Shon[6,*]**

[1]Graduate School of Engineering Science and Technology, National Yunlin University of Science and Technology, Yunlin, 64002, Taiwan

[2]Future Technology Research Center, National Yunlin University of Science and Technology, Yunlin, 64002, Taiwan

[3]Department of Computer Science, Lahore Garrison University, Lahore, 54920, Pakistan

[4]Department of Computer Science and Information Technology, College of Engineering, Abu Dhabi University, Abu Dhabi, 69911, United Arab Emirates

[5]Department of Software Engineering, Faculty of Engineering and Architecture, Nisantasi University, Istanbul, 34398, Turkey

[6]Department of Cybersecurity, Ajou University, Suwon, 16499, Republic of Korea

*Corresponding Author: Taeshik Shon. Email: tsshon@ajou.ac.kr

Received: 01 April 2024 Accepted: 20 June 2024 Published: 18 July 2024

**ABSTRACT**

Cookies are considered a fundamental means of web application services for authenticating various Hypertext Transfer Protocol (HTTP) requests and maintains the states of clients' information over the Internet. HTTP cookies are exploited to carry client patterns observed by a website. These client patterns facilitate the particular client's future visit to the corresponding website. However, security and privacy are the primary concerns owing to the value of information over public channels and the storage of client information on the browser. Several protocols have been introduced that maintain HTTP cookies, but many of those fail to achieve the required security, or require a lot of resource overheads. In this article, we have introduced a lightweight Elliptic Curve Cryptographic (ECC) based protocol for authenticating client and server transactions to maintain the privacy and security of HTTP cookies. Our proposed protocol uses a secret key embedded within a cookie. The proposed protocol is more efficient and lightweight than related protocols because of its reduced computation, storage, and communication costs. Moreover, the analysis presented in this paper confirms that proposed protocol resists various known attacks.

**KEYWORDS**

Cookies; authentication protocol; impersonation attack; ECC

## 1 Introduction

With the advent of state-of-the-art technology, the use of the Internet to access cloud services, online shopping, and social networking sites is progressively becoming an everyday activity among people. When a client visits a particular website for the first time, the website sends a cookie file along with a unique client identifier and stores it in the client's system. In order to obtain information about the client without requiring the re-entry of the same information, whenever the client next visits the same website, the information of the client can be accessed using the stored cookie.

The website's cookies operate in such a manner that they can be read by two methods. Firstly, cookies are tied to HTTP requests and are recognized through the use of cookie headers. Secondly, they can be explicitly requested through an Application Programming Interface (API) call by JavaScript and sent to the server [1]. The cookies request-response mechanism amid the web client and server is defined in the Fig. 1.

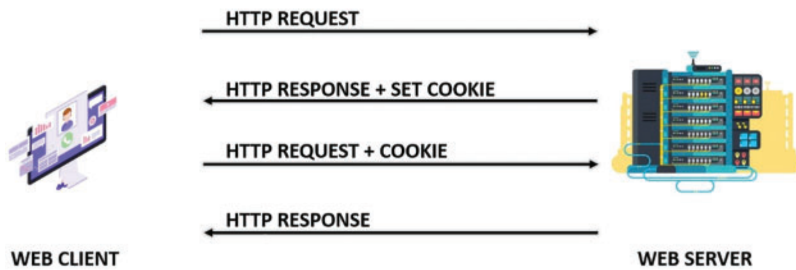


**Figure 1:** HTTP cookies request-response mechanism between web client and server

Conventionally, a cookie consists of numerous attributes, including value, name, path, expiration, and domain. The value attribute is used to store a client's personal information, such as the user's session ID, email address, and identification. The other attributes store unique information that can be used to create customized web pages in the user's browser. This unique information includes products added to the cart or browsing history of a client on a shopping website. Moreover, web cookies are used for the following: (a) cloud services, (b) saved shopping carts, (c) automatic logins, and (d) customized web pages. Furthermore, to evade a hot-linking attack [2]. A session mechanism can also be used with cookies. As the client excessively uses HTTP cookies. Cookies are transmitted across the Internet without any security structure. An adversary can easily access the confidential information stored in the "value" parameter through various cookie-stealing mechanisms.

Microsoft has identified existing security defects in Internet Explorer, where an adversary can steal confidential information from cookies stored in the browser. Likewise, attacks (e.g., Cross-Site Scripting (XSS)) can be launched to send malicious scripts to exploit the client's web browser. These malicious scripts can access any data in cookies stored by a client's browser, and this information can be used on the associated website. Additionally, cookies can be altered on some browsers, such as Mozilla Firefox, through malicious websites. It is possible to modify the cookie's parameters for malicious objectives. For instance, the login session can be extended by modifying the expiration-time parameter. The aforementioned security attacks highlight the problems of both the privacy and integrity of cookies.

Although cookies are widely used by users, their hazardous nature has become a significant concern. Recently, Microsoft has also observed a security deficiency in its Internet Explorer browser, where an adversary can easily steal personal information from stored web cookies. Since the Internet is a public communication channel, stored data can be easily accessed through eavesdropping. Any of the cookie's parameters can be tampered with for malicious purposes, such as using the session expiration parameter to extend the duration of the login session. The above-mentioned threats to security are caused by either of the following problems:

- Cookie Confidentiality: In a cookie, personal information is exposed to eavesdropping whenever it is transmitted in plain text over the Internet. To ensure the privacy of the client, the value of the content, except the server, should not be exposed to anyone.
- Cookie Integrity: Browsers store cookies, and these cookies are transmitted over the Internet without any security features. Hence, to manipulate websites, clients, or servers, these cookies are vulnerable to severe alteration.

The superiority of our proposed method is explained through comprehensive benchmarks demonstrating enhanced security and efficiency compared to existing solutions. The need for this method arises from its unique capability to secure HTTP cookies against evolving cybersecurity threats, thereby providing a robust solution where traditional protocols fall short.

### *1.1 Motivation and Contributions*

The communication between a web client and server over public channels exposes sensitive data to various security threats, necessitating a robust security framework for cookie storage infrastructure. A critical review of existing authentication protocols reveals significant security flaws, particularly in areas of confidentiality, integrity, and resistance to common cyberattacks such as session hijacking and XSS. To address these vulnerabilities, we introduce a secure and lightweight authentication protocol based on ECC.

Our primary contributions to this research work are as follows:

1. We propose an ECC-based protocol that authenticates client-server transactions securely, leveraging the computational efficiency and lower resource requirements of ECC.
2. Our protocol enhances the privacy of HTTP cookies by maintaining strict confidentiality and integrity, setting a new standard in secure communication.
3. It is designed to resist the major security attacks that have compromised previous protocols, offering substantial security benefits and remaining robust yet lightweight.
4. Performance comparisons with existing protocols demonstrate that our proposed solution achieves significant reductions in communication, computation, and storage costs, thereby addressing efficiency concerns effectively.

This work not only mitigates known vulnerabilities but also introduces innovative features that differentiate our protocol from existing solutions, making it a pioneering approach in the field of web security.

## 2 Related Work

In this section, we review the related protocols for cookies. We investigate various cookie protocols, and after analyzing their flaws, we presented an efficient cookie protocol. There are significant limitations are shown in the protocol [3]. Firstly, high-level confidentiality is not offered in their protocol. Second, security against cookie replay attacks are not presented in their protocol. Third, their protocol does not use any procedure for key updating. A cookie protocol is presented by [4] in which a set of inter-dependent cookies are used, e.g., a password cookie, name cookie, life cookie, and a sealed cookie. This protocol does not offer the approach for the confidentiality of cookies.

A survey on web tracking was conducted with cookies by [5]. The information and functionality leaked to adversaries who intercept users' cookies are scrutinized by [6]. In 2018, a side-channel attack was presented [7] against HTTPS that worked by injecting cookies. These studies illustrate the significance of avoiding injecting attacks and cookie hijacking. Various studies have examined security problems related to cookies [8,9]. Cookie confidentiality is not offered by protocol and cookie integrity is also not provided by the protocols [10,11], and only integrity and confidentiality are discussed in the protocols [12–14].

After scrutinizing the protocols, we noticed a common problem the integrity of cookies is not verified by browsers before users start browsing the Internet. Internet protocol based communication methodologies are yet considered to be the most critical selection for setting up the Internet of Things environment [15–17] and SG's networks covering buildings, homes, and more prominent neighborhoods also. The selection of Internet protocols based Smart Grid communications that every smart appliance like television sets, dishwashers, heaters, air conditioners, etc., and smart meter have their own IP addresses and help in quality Internet Engineering Task Force (IETF) schemes for remote management.

The application of power system security using bidirectional RNN-based network anomalous attack detection in cyber-physical systems. The relevance of our cookies security discussion as it highlights the use of advanced security techniques to protect critical infrastructure. Similarly, our cookies security protocol employs advanced methods to ensure the integrity and security of cookies, which are crucial in maintaining the security of web sessions in internet communications [18].

The studies of [19] highlight the importance of anomaly detection in securing communication systems, which is directly applicable to our cookies security protocol. However, reference [20] emphasizes the necessity of authorizing only legitimate communications, a principle that underpins our approach to ensuring the integrity and security of cookies. By employing advanced methods to detect anomalies and authorize communications, our protocol aims to mitigate potential cyber threats, ensuring a secure browsing experience for users.

However, already developed IP-based communication systems, e.g., the Internet, are distinctly possible problems by controlling information and notable delay-sensitive data, and also a wide range of possible malicious attacks, like denial of service attacks, replay attacks, and traffic analysis. So, Internet Protocol (IP) based Smart Grid communications will also be considered vulnerable to security problems. As a consequence, it is necessary to develop Smart Grid communication protocols properly to control all possible security threats. Additionally, not all entities may be trusted in Smart Grid communication. It is required for Smart Grid communication that the entities participating in communication are authenticated whether they are verified and exact if SG communication is utilizing IP-based protocols [21].

Finally, as a resultant, the SG communication framework would be considered an adequate verification mechanism [22–26] so that malicious client is might not able to compromise the privacy or secrecy [27–31] of the information sharing amid the supplier and client [32,33]. Current technologies in Content Delivery Networks (CDN) [34] and smart meters like Advanced Metering Infrastructures (AMI) lead to secrecy concerns because they rely upon centralizing consumption information of the client at smart meters. According to the Netherlander ruling, they concern about the privacy of mobile computing [35–38], fog computing [39], and smart meters [40].

Chachra et al. [41] discuss how affiliate marketing networks provide a structure that connects independent marketers seeking compensation with merchants looking for customers. This interaction occurs when a client visits a site and the browser sends a request containing a cookie to the affiliate network via a tracking pixel. Should the client then purchase goods, the merchant compensates the affiliate network, which in turn pays the independent marketer. Adversaries exploit this mechanism by inserting their own cookies into clients' browsers—a tactic known as cookie stuffing. This fraudulent activity diverts revenue intended for legitimate marketers. The paper provides a measurement-based classification of these cookie-filling scams in online marketing, analyzing the types of affiliates and networks targeted, and the specific fraud tactics employed. It also notes that larger networks are more frequently targeted than smaller, merchant-run affiliate programs. The methodology outlined in the paper is designed to meticulously analyze and measure the performance and operational strategies of large affiliate programs such as Rakuten LinkShare, ShareASale, HostGator Affiliate Program, Amazon Associates Program, CJ Affiliate, and ClickBank. Our approach involves a systematic identification process starting with the targeted merchant, moving through the affiliate network, and down to the specific affiliate ID.

The initial step in our methodology is the identification of the cookies and URLs used by affiliates. This is achieved by gathering online information and, where necessary, by registering with the affiliate programs to gain firsthand data. Each Publisher ID is uniquely linked to an Affiliate ID, facilitating a clear and organized tracking system. Further refining our tracking process, we utilize Google Chrome's extension, Afftracker. This tool enhances our ability to accurately track and associate each Affiliate ID with the domain of the corresponding merchant. By doing so, we can effectively dissect and understand the flow of traffic and the attribution of sales to respective affiliates. This methodical approach not only helps in pinpointing the performance metrics of each affiliate program but also aids in understanding the dynamics between merchants and affiliates, providing a comprehensive overview of affiliate marketing practices across different platforms.

In light of the prevalence of affiliate marketing and the potential risks associated with it, such as cookie stuffing fraud, the above-proposed strategy, based on the Secure Key Exchange Protocol for HTTP Cookies, was carefully examined for its confidentiality and proficiency. By conducting a comparative analysis with existing literature and techniques from relevant investigations, our study sought to address the pressing need for enhanced security measures in cookie management. The results of our investigation revealed that the proposed cookies protocol not only mitigates the risks associated with fraudulent activities such as cookie stuffing but also significantly improves the overall security and effectiveness of cookie handling in web browsing environments. These findings underscore the importance and superiority of our proposed method in comparison to existing approaches, highlighting its potential to provide robust protection against evolving threats in online advertising.

## 3 Preliminaries

In the current section, we explained the notation table and basics of cryptography, such as hash function, ECC, ECDLP, and CDHP. Furthermore, the adversarial model is described to know the abilities of the $\mathtt{A}$.

### *3.1 Elliptic Curve Cryptography (ECC)*

The basics related to ECC used throughout the research are illustrated in this subsection. ECC is based on any chosen real elliptic curve such as $E_p(a, b): y^2 = x^3 + ax + b \bmod p$. Whereas, $a, b \in Z_p$ and $4a^3 + 27b^3 \; mod \; p \neq 0$ for any large prime number $p$. The curve is defined by the integers $(a, b)$. The points $(x, y)$ over $E_p(a, b)$ should verify the previous ECC equations. Repetitive addition is achieved through scalar multiplication defined as $uV = V + V + V + V + V + \ldots + V(u \text{ times})$, where $V$ is a point over $E_p(a, b)$ also $u \in F_p$. Moreover, the same level of security is provided by ECC as compared to traditional key cryptography such as DSA and RSA with smaller key size [42].

#### *3.1.1 Discrete Logarithm Problem Aimed at Elliptic Curve (ECDLP)*

Two specific random points $V, X \epsilon E_p(a, b)$, calculate a scalar $u$ such that $V = uX$. The chances of A that he can compute $u$ in $t$ (polynomial time) is stated as: $ADV_A^{Hash}(\mathrm{t}) = Prb\left[A\,(V, X) = x \colon x\epsilon Z_p\right]$. The assumptions of ECDLP states that $ADV_A^{ECDLP}(\mathrm{t}) \leq \epsilon$.

#### *3.1.2 Computational Diffie-Hellman Problem (CDHP)*

Let $C$ be a cyclic group of order $p$ with generator $c$ and two arbitrary numbers $\alpha, \beta\epsilon Z_p^*$. Computationally it is absurd to compute $\mathrm{c}^{\alpha\beta}$ on the input $(c, c^{\alpha}, c^{\beta})$. In other words, an attacker A has advantage $\delta$ in solving the Computational Diffie-Hellman Problem (CDHP) in ($C$, $p$, $c$) if: $Pr\left[A\,(c, c^{\alpha}, c^{\beta}) = c^{\alpha\beta} \geq \delta\right]$ where the probability is taken over the arbitrary choices, $\beta\epsilon Z_p^*$ and number of bits consumed by the attacker A.

### *3.2 Hash Function*

A deterministic mathematical technique known as a Collison-Resistant one-way hash function, or $h\colon\ (0, 1)^* \rightarrow Z_p^*$, takes variable length inputs and creates fixed length outputs, such as $b$ bits. The term $ADV_A^{Hash}$ (rt) refers to an adversary's advantage in locating a hash collision in run time $rt$. Then $ADV_A^{Hash}\,(\mathrm{rt}) = Prb\left[(k_1, k_2)\,\epsilon Z_p A \colon\ (k1) = h\,(k_1) \neq h\,(k_2)\,, h\,(k_1) = hk_2\right]$, where the probability of random event $X$ is $Prb$[X], and the the pairs $(k_1, k_2)\,\epsilon Z_p$, indicates that the input $k_1$ and $k_2$ are randomly chosen by A. An $(\epsilon, rt)$-adversary A attacking the collision resistance of $h(.)$ means that the run time of A is at most $rt$ and that $ADV_A^{Hash}\,(rt) \leq \epsilon$.

### *3.3 Adversarial Model*

In this subsection, we present the adversarial model as defined in [43], capabilities of the A, based on protocol security definition are as follows:

1. During communication between entities, A has full access to the communication channel (public channel).
2. A can intercept, modify and replay the message or information sent on the communication channel.
3. A can be a legal client on the network.
4. The dynamic identity of the client can be extracted by A.
5. Server is considered secure and A cannot extract server's private key.
6. A can find out previous shared session keys.

## 4 Proposed Protocol

This section provides a detailed description of our proposed protocol based on ECC. Where a client sends a pseudo-identity to the server to be registered himself, the server sends a message with parameters for completion of registration. After completion of the registration process, the client sends a login request message. Receiving a request message, a server transmits parameters with a challenge request message, and after that, when all authentication gets completed, the session key is shared to start the services between server and client. An operational procedure and comparison with other related protocols are also provided.

The proposed protocol consists of three major phases as described in below subsections. Primarily, we used both random numbers and time stamps for protection against several attacks. The notations are listed in Table 1 and description and analysis of the proposed protocol are presented in Proposed Protocol.

**Table 1:** Notation table

| Notations | Description |
|---|---|
| $ID_u$ | Identity of client |
| $S$ | Secret key of server |
| $P$ | Base point of the elliptic curve $E_p(a, b)$ |
| $Cu_1$ | Non sensitive information |
| $Cu_2$ | Sensitive information |
| $h(.)$ | Hash functions |
| $SK$ | Shared session key between client and server |
| $\|$ | Concatenation |
| $\oplus$ | XoR operation |
| $\mathbb{A}$ | An adversary |
| $a_u$, $r_u$ | Random numbers of client |
| $E$, $B$, $C$ | Variables |
| $T_1$, $T_2$, $T_3$, $T_4$ | Current time stamps |

| *Client* | | Server |
|---|---|---|
| **Registration Phase** | | |
| Selects $ID_u$<br>Generate a random number $r_u$<br>and compute $PID_u = \{h(ID_u \| r_u\}$ | | |
| | $\xrightarrow{\{ID_u, PID_u\}}$ | |
| | | Stores $PID_u$ in its database<br>Compute $A_i = h(PID \| s)$<br>Generate $P_k = sp$ |
| | $\xleftarrow{\{issues\ A_j, P_k\}}$ | |
| Client stores $A_i$ and $P_k$ for further uses | | |
| **Login and Authentication Phase** | | |
| Selects $a_u$ $B = a_u P_k = a_u sP$<br>$C = h(A_i \| a_u P \| T1)$ | | |
| | $\xrightarrow{\{PID_u, B, C, T_1\}}$ | |
| | | Check time stamp $T_2 - T_1 \le \Delta T$<br>abort if not equal<br>$a_u P = s^{-1} B$<br>$C \stackrel{?}{=} (h(PID_u \| s) \| a_u P \| T_1)$<br>Session aborts, if above equation not verified<br>Generates $C_{u1}$ and $C_{u2}$<br>$E = a_u P \oplus C_{u2}$ |

(Continued)

**Table 1 (continued)**

| Notations | Description | |
|---|---|---|
| | $\overleftarrow{\{E, C_{u1}, \mathrm{T}_3)\}}$ | $SK = (h((PID_u \| s) \| a_u P \| T_3$ |
| Check time stamp $T_4 - T_3 \leq \Delta T$ abort if not fresh | | |
| $C_{u2} = a_u P \oplus E$ | | |
| $SK \stackrel{?}{=} (A_i \| a_u P \| C_{u2} \| T_2)$ | | |
| | $\longleftarrow$ $Sk = Sk$ $\longrightarrow$ | |

A detailed description of the above phases is given as follows.

### *4.1 Registration Phase*

In this section, we present the client registration process with the server. Following steps are executed, once a client initiates a registration request:

REG Step 1: The client selects an identity $ID_u$ that will be unique to get services from the server, generates a random client number $r_u$ and computes $PID_u = \{h(ID_u || r_u\}$ where $PID_u$ is the pseudo-identity of a client that is generated by concatenation of identity and random number $r_u$ of the client which is protected with one-way hash function to make secure in order to make client's identity anonymous. Then the client sends both $ID_u$ and $PID_u$ over a secure channel to get himself registered with the server to get services from the server.

REG Step 2: Server stores the $PID_u$ in its database for the later usage and computes $A_i = h(PID||s)$ where concatenation of $PID_u$ and secret key $s$ by hash function.

REG Step 3: After calculation of $A_i$, the secret key multiply with a large prime number and make a copy as $A_i = sP$; at the end of the registration phase, the server returns a pair $(A_i, P_k)$ to the client and the client keeps this pair for further usage.

### *4.2 Login and Authentication Phase*

This section presents the login and authentication phase of the proposed scheme, which is also summarized in Proposed Protocol.

LA Step 1:

The client selects a random number $a_u$ and calculates an equation $B = a_u P_k = a_u sP$ for $s$. Furthermore, stored parameters $A_i$, $a_u P_k$ and time stamp $T_1$ in the client's database, he computes an equation in the following manner $C = h(A_i| |a_u P| |T1)$. After the calculation of the above equation, the client transmits the request message containing $PID_u$, $B$, $C$, and $T_1$ to the server for login over a public channel.

LA Step 2:

After the successful receiving of message request containing $\{PID_u, B, C, T_1\}$, the server checks the time stamp $T_2 - T_1 \leq \Delta T$ to check the freshness of message. The server calculates $a_u P = s^{-1} B$. Otherwise, if the time stamp is not fresh, the session will be abandoned. Then server computes and verify $C \stackrel{?}{=} h(h(PID_u \| s) \| a_u P \| T_1)$. If this verification is not authenticated, then the session will be aborted right here; otherwise, the server generates cookies $C_{u1}$ and $C_{u2}$. Where, $C_{u1}$ is non sensitive information and $C_{u2}$ is sensitive information. Then calculates the equation $E = a_uP \oplus C_{u2}$ that becomes an unknown value. After that, creates a session key $SK$ through the equation $SK = (h((PID_u||s)||a_uP||T_3$. Calculated parameters $E$, $C_{u1}$, and $T_3$ send to the client so that he can check whether the server is trusted or not.

LA Step 3:

In order to authenticate the receiving challenge message from the server containing $E$, $C_{u1}$, $T_2$ the client checks the time stamp $T_4 - T_3 \leq \Delta T$. Client computes $C_{u2} = a_uP \oplus E$. Otherwise, the session time will be aborted if the time stamp is not fresh. After calculations of the above values, the client gets checks the session key $SK \stackrel{?}{=} h\,(h(PID_u||s)\,||C_{u2}||a_uP||T_3)$. This procedure outlines the method by which a session key is securely shared between the client and server. Once the session key is established and mutual authentication is confirmed, the client is authorized to access services provided by the server.

## 5 Security Analysis

In Section 5, we provide a quick overview of both formal and informal security evaluations. These introductory remarks lay the groundwork for the full examination of the security characteristics and effectiveness of our suggested protocol in the following subsections.

### *5.1 Information Security Analysis*

The correctness and security of the proposed scheme are shown in the current section. Analysis of this scheme shows its robustness, improving the effectiveness of security and defense from different kinds of attacks, which are discussed in given below.

#### *5.1.1 Ensuring of Mutual Authentication*

The mutual authentication between client and server is ensured as following steps. The server authenticates the client by checking $C \stackrel{?}{=} h\,(h(PID_u||s)\,|\,|a_uP|\,|T_1).h(PID_u||s)$ and $a_uP$ are needed to calculate $C$ successfully by $\mathtt{A}$. The computation of $h(PID_u||s)$ and $a_uP$ imply the secret key $s$ of the server, which is not known by $\mathtt{A}$. So, only the legal server can authenticate the client. Likewise, the client authenticates server by computing $SK \stackrel{?}{=} A_i||a_uP||C_{u2}||T_2)$, $\mathtt{A}$ needs to calculate the $A_i$ to get access but it requires secret key $s$ of server. Furthermore, adversary is unable to compute $C_{u2}$.

#### *5.1.2 Providing Client Anonymity*

Anonymity and privacy are considered significant features during making an authentication protocol. If anonymity is revealed to any $\mathtt{A}$, the client's information, like location, social circle, moving history, and priorities, can be accessed by $\mathtt{A}$. In the registration phase, the client calculates $PID_u = \{h(ID_u||r_u)\}$ applying a hash function on the concatenated values of a random number $r_u$ and $ID_u$. The pseudo-identity $PID_u$ of the client is transmitted to legal sever instead of $PID_u$ in login message $PID_u$, $B$, $C$, $T_1$. Each successful authentication session executes a new pseudo-identity, $PID_u$. Additionally, the client generates a session-specific random integer $a_u$ that prevents an adversary from determining if two independent sessions are initiated by the same or separate clients. Therefore, our protocol makes each client's privacy and anonymity possible.

The conditions of anonymity:

(i) The identity of the client should not be leaked.
(ii) It should not determine that the same client initiated two different sessions.

So, both conditions of anonymity are fulfilled in this protocol. This protocol ensured the anonymity of the client.

### *5.1.3 Defense against Client Impersonation Attack*

If an adversary $\mathtt{A}$ wants to impersonate a legal client, then he must have to issue an authentic and valid login request message $\{PID_u, B, C, T_1\}$. So, for the calculation of $PID_u = \{h(ID_u||r_u)\}$, $\mathtt{A}$ requires client's identity. Similarly, for the calculation of $C = h(A_i||a_uP||T_1)$, $\mathtt{A}$ requires the correct value of $A_i = h(PID||s)$ which is possible to compute by having the private key of the server. Because the identity and secret key of the server are not known to $\mathtt{A}$, our protocol can be considered more secure for defense against client impersonation attacks.

### *5.1.4 Defense against Server Impersonation attack*

If $\mathtt{A}$ desires to impersonate an authentic server, then he must have to generate an authentic challenge message $\{E, c_{u2}, T_3\}$ }. For calculation of $E = a_uP \oplus C_{u2}$, it requires $a_uP = s^{-1}B$, which is possible to compute by having the private key of the server. So, it is clear that our proposed protocol is secured against server impersonation attacks.

### *5.1.5 Defense against Man-in-Middle Attack*

If $\mathtt{A}$ can calculate the authentication restriction between client and server, and the man-in-middle attack will be possible. If $\mathtt{A}$ has values $B$ and $C$, he can be able to pass an authentication check. Similarly, he can also pass an authentication check of the legal server if $\mathtt{A}$ contains the server's secret key $s$. Due to the above checks, $\mathtt{A}$ cannot get all the mentioned calculations, so authentication checks cannot be passed. So, the proposed protocol facilitates the feature against man-in-middle attacks.

### *5.1.6 Providing Perfect Forward Secrecy*

Perfect forward secrecy is an important need for designing an authentication protocol. It makes assure that the secrecy of already used previous session keys remains secure in case a long-term private key, password, or session key of any participant is revealed. In our presented protocol, every shared key $SK = (h((PID_u||s)||a_uP||T_3)$ contains the session specific random number au produced by server. Similarly, $SK \stackrel{?}{=} A_i||a_uP||C_{u2}||T_2)$ contains the session specific random number au produced by the client. So, if a shared or long-term private key is revealed, already-used session keys cannot be compromised.

## ***5.2 Formal Security Analysis***

In this subsection, the proposed protocol is evaluated formally using the random oracle model: Security Proof: In order to understand the security strength of our protocol, two types of security requirements, like, integrity and authentication based on the Random Oracle Model (ROM), are discussed here. For this purpose, the following definitions are considered:

**Security Proof:** $\mathtt{A}$ is a person who is not registered with a system. But, $\mathtt{A}$ has knowledge of all the messages which are being transmitted over a public channel.

**Theorem T1:** Authentication property under the assumption of a hash function is being satisfied.

**Proof:** In order to get access to the system, the client must enter values like $ID_u$ and random number $r_u$ as per the presented protocol. $ID_u$ can be known by $\mathtt{A}$ easily but he is unable to know the random number $r_u$, because it is only known by client. At the time of login, client inserts $a_u$ and computes:

$$B = a_u P_k = a_u sP \quad (1)$$

$$C = h(A_i|\,|a_u P|\,|T1) \quad (2)$$

Furthermore, upon receiving the challenge message, the subsequent value is computed:

$$C_{u2} = a_u P \oplus E \quad (3)$$

and check $SK \stackrel{?}{=} A_i||a_u P||C_{u2}||T_2)$ is performed to determine the client's legitimacy. This check will be passed only if the client has inserted valid credentials. Moreover, there is no way for $\mathtt{A}$ to know the secret parameters of the client.

**Theorem T2:** The proposed protocol is secured against integrity attacks under a secure hash function in ROM with polynomial time.

**Proof:** Integrity property of all transmitted messages must be satisfied to prove the correctness of the message. In our proposed protocol, the client transmits message $\{PID_u, B, C, T_1\}$ to the server over a public channel. So, $\mathtt{A}$ can try to intercept and modify the message $\{PID_u, B, C, T_1\}$ In order to deal with this issue and to maintain the integrity of the message, the concept of a secure hash function is used. Whereas the secure hash function is an irreversible function. On the server side, the server computes the following:

$$a_u P = s^{-1} B \quad (4)$$

and determines $C \stackrel{?}{=} h\,(h(PID_u||s)\,|\,|a_u P|\,|T_1)$ to confirm the integrity of the message received from the client. If this condition holds, then it means that the received message is correct and not modified, but if this condition fails, then it means that the message is intercepted and modified by $\mathtt{A}$. In this case, the server discards the message immediately. So, this is the way the receiver can guess the correctness of the message transmitted over a public channel. Thus, the proposed protocol is secured against integrity attacks.

## 6 Performance Analysis

In this section, we state the performance of the proposed protocol. The explanation and implementation of the proposed and related protocols are given below:

Cryptographic-operations $(T_{SM}, T_{OWH}, T_{AE}, T_{PA}, T_{PM}, T_{SE}, T_{SD}, T_{HMAC}, T_{AD}, T_{\oplus}, T_{s||})$ are implemented in Ubuntu utilizing PyCrypto library, with an 8.0 GB RAM and 2.60 GHZ processor with core i7 using Python programming-language. This verification protocol executed 10 times with the same suppositions by average time. Operations $(T_{\oplus})$ and $(T_{s||})$ take less execution time. So, these operations are not included in the computations of total time. The operation $T_{OWH}(.)$ takes 0.00070 ms for execution while $T_{pm}$ takes 0.0020 ms for point multiplication. The running time of cryptographic operations is described in Table 2.

**Table 2:** Time for cryptographic operations

| Notation | Description | Required time in ms |
|---|---|---|
| $T_{SM}$ | Exhibits running time for ECC scalar multiplication | 0.0240 |
| $T_{owh}$ | Exhibits running time for one-way hash function | 0.00070 |

(Continued)

**Table 2 (continued)**

| Notation | Description | Required time in ms |
|---|---|---|
| $T_{AE}$ | Exhibits running time for modular exponentiation | 0.0040 |
| $T_{PA}$ | Exhibits running time for point addition | 0.0030 |
| $T_{PM}$ | Exhibits running time for point multiplication | 0.00201 |
| $T_{SE}$ | Exhibits running time for symmetric key encryption | 0.0250 |
| $T_{SD}$ | Exhibits running time for symmetric key decryption | 0.0100 |
| $T_{HMAC}$ | Exhibits running time for hash-based message authentication code | 0.0341 |
| $T_{AD}$ | Exhibits running time for asymmetric key decryption | 0.0025 |

Moreover, Tables 3 and 4 present computational, storage, and communication costs of the proposed protocol in contrast to relevant protocols [44–48] as follows.

**Table 3:** Aggregated computation cost

| Protocol | Computation cost |
|---|---|
| Proposed work | $6T_{owh(\cdot)} + 9T_{pm} = 0.00222$ ms |
| Mahmood et al. [44] | $5T_{SM} + 5T_{owh(\cdot)} + 1T_{PA} = 0.1265$ ms |
| Wazid et al. [45] | $26T_{owh(\cdot)} + 4T_{SM} = 0.1142$ ms |
| Eftikhari et al. [46] | $26T_{owh(\cdot)} + 6T_{SM} + 3T_{PA} = 0.1712$ ms |
| Wu et al. [47] | $21T_{owh(\cdot)} + 6T_{SM} = 0.1587$ ms |
| Chen et al. [48] | $19T_{owh(\cdot)} = 0.0133$ ms |

**Table 4:** Aggregated communication and storage cost

| Protocol | Communication cost | Storage cost |
|---|---|---|
| Proposed work | 1312 bits | 672 bits |
| Mahmood et al. [44] | 1600 bits | 320 bits |
| Wazid et al. [45] | 3392 bits | 1536 bits |
| Eftikhari et al. [46] | 4704 bits | 768 bits |
| Wu et al. [47] | 5376 bits | 832 bits |
| Chen et al. [48] | 2208 bits | 928 bits |

### *6.1 Comparisons of Communication Cost*

Fig. 2 refers to the comparison summary of aggregated calculated communication costs between relevant and proposed protocols.

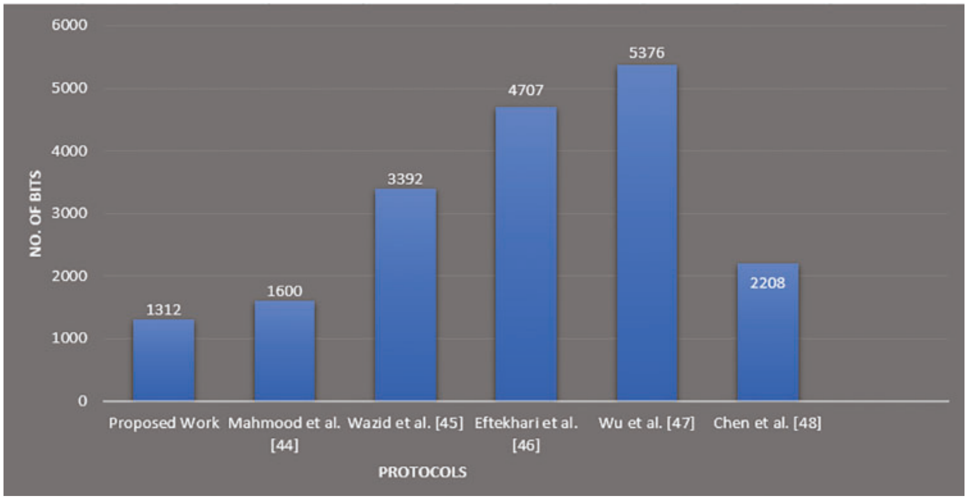


**Figure 2:** Comparisons of communication cost between proposed and related protocols

The reserved bits are considered for timestamps, identity, point addition, and point multiplication are specified as 160 bits, encryption/decryption 128 bits, and hash takes 256 bits. Based on these assumptions, it is observed that calculations are presented in Table 4 for the sake of storage and calculation cost for proposed and relevant protocol [44–48]. It presents the trade-off between performance and confidentiality, whilst the proposed protocol proposes extra-aided confidentiality features.

### *6.2 Comparisons of Computation Cost*

The comparison summary between related and proposed protocol The computation cost is presented in the Fig. 3 and is depicted in Table 3 as well. The list of relevant and proposed protocols is marked vertically, while the required time in milliseconds for computation is marked horizontally in the graph. It is observed easily that the proposed protocol takes less time than a few relevant protocols for analysis.

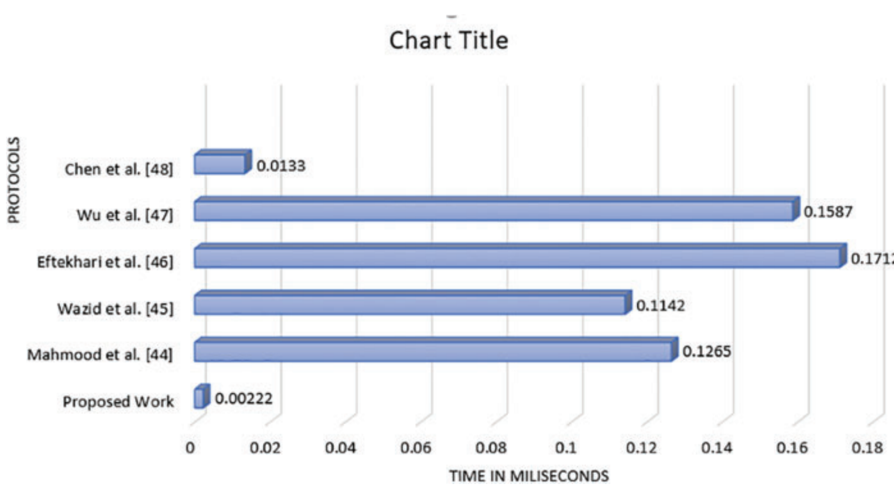


**Figure 3:** Comparisons of computation cost between proposed and related protocols

### *6.3 Comparisons of Storage Cost*

The storage costs for both related and proposed protocols are systematically compared in Fig. 4 and Table 4.

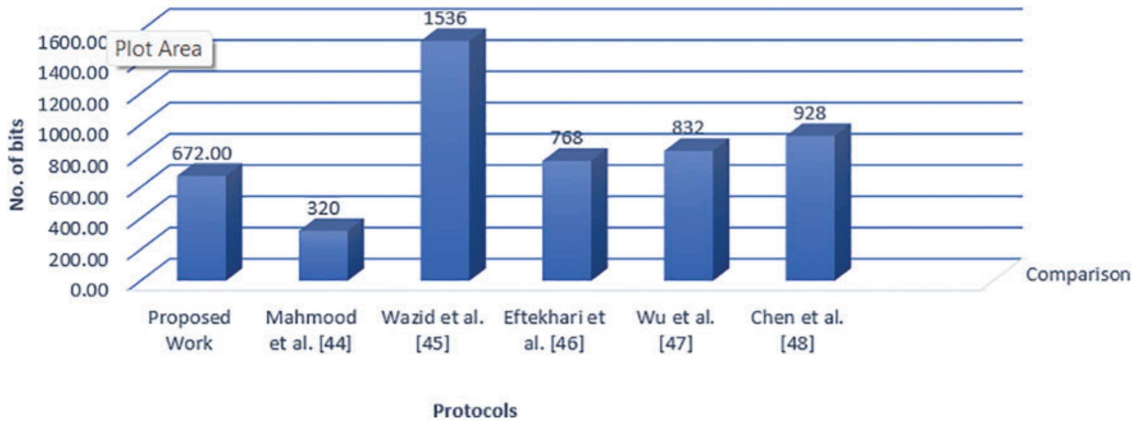


**Figure 4:** Comparisons of storage cost between proposed and related protocols

The graph in Fig. 4 displays the required bits for storage vertically, with the related and proposed protocols labeled horizontally. Notably, the proposed protocol allocates more bits for storage compared to various relevant protocols. This increased storage requirement stems from its advanced confidentiality features, which enhance the overall security of the protocol.

Upon detailed analysis of the data presented in Tables 3–5, it becomes clear that the communication, computation, and storage costs associated with our protocol are substantially lower than those incurred by many existing protocols in the field. This indicates a significant improvement in efficiency and resource management. Additionally, the proposed protocol not only meets standard security requirements but also introduces advanced security features that provide superior protection and robustness compared to other protocols that address similar issues.

**Table 5:** Confidentiality features: Comparison summary between proposed and relevant protocols

| Protocol→Security features↓ | Proposed | Mahmood et al. [44] | Wazid et al. [45] | Eftikhari et al. [46] | Wu et al. [47] | Chen et al. [48] |
|---|---|---|---|---|---|---|
| Impersonation attack | Yes | Yes | No | Yes | Yes | Yes |
| Replay attack | Yes | Yes | Yes | Yes | Yes | Yes |
| Client anonymity | Yes | No | Yes | Yes | Yes | Yes |
| Perfect forward secrecy | Yes | Yes | Yes | No | Yes | Yes |
| Man in middle attack | Yes | Yes | No | Yes | No | Yes |
| Mutual authentication and key agreement | Yes | Yes | Yes | Yes | Yes | Yes |
| Denial of service attack | Yes | Yes | Yes | Yes | Yes | No |

This enhanced security aspect makes our protocol a more reliable and attractive option for deployment in environments requiring stringent security measures.

## 7 Conclusion and Future Directions

Our conclusion has been enhanced to better summarize the key findings, including the identification of various issues such as cost, privacy, and security challenges in cookie management and online transactions. We introduced an ECC-based lightweight, secure, and efficient key agreement authentication protocol designed to tackle these problems through secure cryptographic operations. Our free study evaluating the security of this protocol and a detailed comparative analysis of computation, communication, and storage costs demonstrate its superior efficiency and security over existing protocols. Additionally, we acknowledge the limitations of our research, particularly in the scalability of the protocol across diverse environments, and recommend future studies to explore this area further.

In the future, we will focus on improving cookie security in affiliate marketing to offer strong protection against unwanted tracking and data breaches. We want to create standards that protect user data while ensuring transparency and compliance in affiliate networks.

**Acknowledgement:** None.

**Funding Statement:** Shehzad Ashraf Chaudhry acknowledges financial support from Abu Dhabi University's Office of Research and Sponsored Programs Grant Number: 19300810.

**Author Contributions:** Study conception, Design, Conceptualization, Data curation, Writing–original draft, Methodology: Waseem Akram; Supervision, Methodology, Writing–original draft, Writing–review & editing: Khalid Mahmood; Resources, Software: Hafiz Burhan ul Haq; Validation, Visualization, Draft manuscript preparation: Muhmmad Asif; Investigation, Supervision, Writing–original draft, Writing–review & editing: Shehzad Ashraf Chaudhry; Formal and informal analysis: Taeshik Shon. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** The authors confirm that the data supporting the findings of this study are available within the article.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

[1] F. Roesner, T. Kohno, and D. Wetherall, "Detecting and defending against third-party tracking on the web," in *9th USENIX Symp. Netw. Syst. Des. Implement. (NSDI 12)*, San Jose, CA, USA, 2012, pp. 155–168.

[2] Z. Chu and H. Wang, "An investigation of hotlinking and its countermeasures," *Comput. Commun.*, vol. 34, no. 4, pp. 577–590, 2011. doi: 10.1016/j.comcom.2010.05.007.

[3] K. Fu, E. Sit, K. Smith, and N. Feamster, "The dos and don'ts of client authentication on the web," in *10th USENIX Secur. Symp. (USENIX Security 01)*, Washington, DC, USA, 2001. Accessed: Jul. 07 2024. [Online]. Available: http://www.usenix.org/events/sec01/fu/fu.pdf

[4] J. S. Park and R. Sandhu, "Secure cookies on the web," *IEEE Internet Comput.*, vol. 4, no. 4, pp. 36–44, 2000. doi: 10.1109/4236.865085.

[5] T. Bujlow, V. Carela-Espanol, J. Sole-Pareta, and P. Barlet-Ros, "A survey on web tracking: Mechanisms, implications, and defenses," *Proc. IEEE*, vol. 105, no. 8, pp. 1476–1510, 2017. doi: 10.1109/JPROC.2016.2637878.

[6] S. Sivakorn, I. Polakis, and A. D. Keromytis, "The cracked cookie jar: HTTP cookie hijacking and the exposure of private information," in *2016 IEEE Symp. Secur. Priv. (SP)*, San Jose, CA, USA, IEEE, 2016, pp. 724–742.

[7] F. Chen, H. Duan, X. Zheng, J. Jiang, and J. Chen, "Path leaks of HTTPS side-channel by cookie injection," in *Construct. Side-Channel Analy. Secur. Design: 9th Int. Works.*, Singapore, Springer International Publishing, 2018, pp. 189–203.

[8] C. Blundo, S. Cimato, and R. D. Prisco, "A lightweight approach to authenticated web caching," in *The 2005 Symp. Appl. Internet*, Trento, Italy, IEEE, 2005, pp. 157–163.

[9] W. B. Shahid, B. Aslam, H. Abbas, S. B. Khalid, and H. Afzal, "An enhanced deep learning based framework for web attacks detection, mitigation and attacker profiling," *J. Netw. Comput. Appl.*, vol. 198, no. 3, pp. 103270, 2022. doi: 10.1016/j.jnca.2021.103270.

[10] A. Juels, M. Jakobsson, and T. N. Jagatic, "Cache cookies for browser authentication," in *2006 IEEE Symp. Secur. Priv. (S&P'06)*, Oakland City, CA, USA, IEEE, 2006, pp. 5.

[11] A. X. Liu, J. M. Kovacs, C. T. Huang, and M. G. Gouda, "A secure cookie protocol," in *Proc. 14th Int. Conf. Comput. Commun. Netw., ICCCN 2005*, San Diego, CA, USA, IEEE, Oct. 17–19, 2005, pp. 333–338.

[12] D. Xu, C. Lu, and A. Dos Santos, "Protecting web usage of credit cards using one-time pad cookie encryption," in *18th Annu. Comput. Secur. Appl. Conf., Proc.*, Las Vegas, NV, USA, IEEE, 2002, pp. 51–58.

[13] J. P. Yang and K. H. Rhee, "A new design for a practical secure cookies system," *J. Inform. Sci. Eng.*, vol. 22, no. 3, pp. 559–571, 2006.

[14] I. Dacosta, S. Chakradeo, M. Ahamad, and P. Traynor, "One-time cookies: Preventing session hijacking attacks with stateless authentication tokens," *ACM Trans. Internet Technol.*, vol. 12, no. 1, pp. 1–24, 2012. doi: 10.1145/2220352.2220353.

[15] S. A. Chaudhry, A. Irshad, K. Yahya, N. Kumar, M. Alazab and Y. B. Zikria, "Rotating behind privacy: An improved lightweight authentication scheme for cloud-based IoT environment," *ACM Trans. Internet Technol.*, vol. 21, no. 3, pp. 1–19, 2021. doi: 10.1145/3425707.

[16] S. A. Chaudhry, M. S. Farash, N. Kumar, and M. H. Alsharif, "PFLUA-DIoT: A pairing free lightweight and unlinkable user access control scheme for distributed IoT environments," *IEEE Syst. J.*, vol. 16, no. 1, pp. 309–316, 2020. doi: 10.1109/JSYST.2020.3036425.

[17] K. Mahmood, W. Akram, A. Shafiq, I. Altaf, M. A. Lodhi and S. K. H. Islam, "An enhanced and provably secure multi-factor authentication scheme for Internet-of-Multimedia-Things environments," *Comput. & Elect. Eng.*, vol. 88, no. 3, pp. 106888, 2020. doi: 10.1016/j.compeleceng.2020.106888.

[18] S. Kwon, H. Yoo, T. Shon, "IEEE 1815.1-based power system security with bidirectional RNN-based network anomalous attack detection for cyber-physical system," *IEEE Access*, vol. 8, pp. 77572–77586, 2020. doi: 10.1109/ACCESS.2020.2989770.

[19] S. J. Kim, W. Y. Jo, T. Shon, "APAD: Autoencoder-based payload anomaly detection for industrial IoE," *Appl. Soft Comput.*, vol. 88, pp. 106017, 2020. doi: 10.1016/j.asoc.2019.106017.

[20] W. Jo, S. J. Kim, H. Kim, Y. Shin, and T. Shon, "Automatic whitelist generation system for ethernet based in-vehicle network," *Comput. Ind.*, vol. 142, no. 1, pp. 103735, 2022. doi: 10.1016/j.compind.2022.103735.

[21] S. A. Chaudhry, "Correcting PALK: Password-based anonymous lightweight key agreement framework for smart grid," *Int. J. Elect. Power & Energy Systems*, vol. 125, no. 3, pp. 106529, 2021. doi: 10.1016/j.ijepes.2020.106529.

[22] H. Zhu, X. Lin, R. Lu, P. H. Ho, and X. Shen, "SLAB: A secure localized authentication and billing scheme for wireless mesh networks," *IEEE Trans. Wirel. Commun.*, vol. 7, no. 10, pp. 3858–3868, 2008. doi: 10.1109/T-WC.2008.07418.

[23] X. Lin, R. Lu, P. H. Ho, X. Shen, and Z. Cao, "TUA: A novel compromise-resilient authentication architecture for wireless mesh networks," *IEEE Trans. Wirel. Commun.*, vol. 7, no. 4, pp. 1389–1399, 2008. doi: 10.1109/TWC.2008.060990.

[24] K. Kursawe, G. Danezis, and M. Kohlweiss, "Privacy-friendly aggregation for the smart-grid," in *Priv. Enhan. Technol.: 11th Int. Symp.*, Waterloo, ON, Canada, Springer Berlin Heidelberg, Jul. 27–29, 2011, pp. 175–191.

[25] S. A. Chaudhry, K. Yahya, S. Garg, G. Kaddoum, M. M. Hassan and Y. B. Zikria, "LAS-SG: An elliptic curve-based lightweight authentication scheme for smart grid environments," *IEEE Trans. Ind. Inform.*, vol. 19, no. 2, pp. 1504–1511, 2022. doi: 10.1109/TII.2022.3158663.

[26] A. R. Metke and R. L. Ekl, "Smart grid security technology," in *2010 Innov. Smart Grid Technol. (ISGT)*, IEEE, 2010, pp. 1–7.

[27] M. Kgwadi and T. Kunz, "Securing RDS broadcast messages for smart grid applications," in *Proc. 6th Int. Wireless Commun. Mobile Comput. Conf.*, Caen, France, Jun. 28–Jul. 2, pp. 1177–1181.

[28] R. Lu, X. Li, X. Liang, X. Shen, and X. Lin, "GRS: The green, reliability, and security of emerging machine to machine communications," *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 28–35, 2011. doi: 10.1109/MCOM.2011.5741143.

[29] G. N. Ericsson, "Cyber security and power system communication—Essential parts of a smart grid infrastructure," *IEEE Trans. Power Deliv.*, vol. 25, no. 3, pp. 1501–1507, 2010. doi: 10.1109/TPWRD.2010.2046654.

[30] S. A. Chaudhry *et al.*, "An anonymous device to device access control based on secure certificate for internet of medical things systems," *Sustain. Cities Soc.*, vol. 75, no. 1, pp. 103322, 2021. doi: 10.1016/j.scs.2021.103322.

[31] A. Hamlyn, H. Cheung, T. Mander, L. Wang, C. Yang and R. Cheung, "Network security management and authentication of actions for smart grids operations," in *2007 IEEE Canada Elect. Power Conf.*, Montreal, QC, Canada, IEEE, Oct. 25, 2007, pp. 31–36.

[32] X. Lin, X. Sun, P. H. Ho, and X. Shen, "GSIS: A secure and privacy-preserving protocol for vehicular communications," *IEEE Trans. Vehicular Technol.*, vol. 56, no. 6, pp. 3442–3456, 2007. doi: 10.1109/TVT.2007.906878.

[33] R. Lu, X. Lin, H. Zhu, P. H. Ho, and X. Shen, "ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications," in *IEEE INFOCOM 2008-The 27th Conf. Comput. Commun.*, Phoenix, AZ, USA, IEEE, Apr. 13, 2008, pp. 1229–1237.

[34] W. Ali, A. Khan, and W. Akram, "Analyzing the deployment and performance of Multi-CDNs in Pakistan," *Pakistan J. Eng. Technol.*, vol. 4, no. 1, pp. 169–174, 2021.

[35] A. Irshad, S. A. Chaudhry, O. A. Alomari, K. Yahya, and N. Kumar, "A novel pairing-free lightweight authentication protocol for mobile cloud computing framework," *IEEE Syst. J.*, vol. 15, no. 3, pp. 3664–3672, 2020. doi: 10.1109/JSYST.2020.2998721.

[36] C. Cuijpers and B. J. Koops, "The 'smart meters' bill: A privacy test based on article 8 of the ECHR," 2008. Accessed: Jul. 07, 2024. [Online]. Available: https://smartgridawareness.org/wp-content/uploads/2014/11/dutch-smart-meters-report-tilt-october-2008-english-version.pdf

[37] Z. Ali, S. A. Chaudhry, K. Mahmood, S. Garg, Z. Lv, and Y. B. Zikria, "A clogging resistant secure authentication scheme for fog computing services," *Comput. Netw.*, vol. 185, no. 9, pp. 107731, 2021. doi: 10.1016/j.comnet.2020.107731.

[38] K. Mahmood, J. Arshad, S. A. Chaudhry, and S. Kumari, "An enhanced anonymous identity-based key agreement protocol for smart grid advanced metering infrastructure," *Int. J. Commun. Syst.*, vol. 32, no. 16, pp. e4137, 2019. doi: 10.1002/dac.4137.

[39] W. Akram, K. Mahmood, X. Li, M. Sadiq, Z. Lv and S. A. Chaudhry, "An energy-efficient and secure identity based RFID authentication scheme for vehicular cloud computing," *Comput. Netw.*, vol. 217, no. 4, pp. 109335, 2022. doi: 10.1016/j.comnet.2022.109335.

[40] S. A. Chaudhry *et al.*, "A lightweight authentication scheme for 6G-IoT enabled maritime transport system," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 2401–2410, 2021. doi: 10.1109/TITS.2021.3134643.

[41] N. Chachra, S. Savage, and G. M. Voelker, "Affiliate crookies: Characterizing affiliate marketing abuse," in *Proc. 2015 Internet Measur. Conf.*, Tokyo, Japan, 2015, pp. 41–47.

[42] S. William, *Cryptography and Network Security: For VTU*. 4th ed. Chennai, India: Pearson Education India, 2006. Accessed: Jul. 07, 2024. [Online]. Available: https://books.google.com.tw/books?id=Pl47qiuv5sgC

[43] H. Lee, J. Nam, M. Kim, and D. Won, "Forward anonymity-preserving secure remote authentication scheme," *KSII Transact. Int. Inform. Syst. (TIIS)*, vol. 10, no. 3, pp. 1289–1310, 2016.

[44] K. Mahmood, S. A. Chaudhry, H. Naqvi, S. Kumari, X. Li and A. K. Sangaiah, "An elliptic curve cryptography based lightweight authentication scheme for smart grid communication," *Future Gener. Comput. Syst.*, vol. 81, no. 2, pp. 557–565, 2018. doi: 10.1016/j.future.2017.05.002.

[45] M. Wazid, P. Bagga, A. K. Das, S. Shetty, J. J. P. C. Rodrigues, and Y. Park, "AKM-IoV: Authenticated key management protocol in fog computing-based Internet of vehicles deployment," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8804–8817, 2019. doi: 10.1109/JIOT.2019.2923611.

[46] S. A. Eftekhari, M. Nikooghadam, and M. Rafighi, "Security-enhanced three-party pairwise secret key agreement protocol for fog-based vehicular ad-hoc communications," *Veh. Commun.*, vol. 28, no. 1, pp. 100306, 2021. doi: 10.1016/j.vehcom.2020.100306.

[47] T. Y. Wu, Z. Lee, L. Yang, J. N. Luo, and R. Tso, "Provably secure authentication key exchange scheme using fog nodes in vehicular ad hoc networks," *J. Supercomput.*, vol. 77, no. 7, pp. 6992–7020, 2021. doi: 10.1007/s11227-020-03548-9.

[48] C. M. Chen, Z. Li, S. Kumari, G. Srivastava, K. Lakshmanna and T. R. Gadekallu, "A provably secure key transfer protocol for the fog-enabled Social Internet of Vehicles based on a confidential computing environment," *Veh. Commun.*, vol. 39, no. 1, pp. 100567, 2023. doi: 10.1016/j.vehcom.2022.100567.