IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 3, 2019, accepted September 4, 2019, date of publication September 11, 2019,
date of current version September 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2940331

Key-Based Cookie-Less Session Management Framework for Application Layer Security

**ZAHOOR AHMED ALIZAI[1], HASAN TAHIR[1], MALIK HAMZA MURTAZA[1],
SHAHZAIB TAHIR[2], (Member, IEEE), AND
KLAUS MCDONALD-MAIER[3], (Senior Member, IEEE)**

[1]School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan
[2]Department of Information Security, College of Signals, National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan
[3]School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K.

Corresponding author: Shahzaib Tahir (shahzaib.tahir@mcs.edu.pk)

This work has been supported by the European CHIST-ERA SPIRIT Project funded in the UK by the Engineering and Physical Sciences Research Council (EPSRC) [grant number EP/P016006/1].

ABSTRACT The goal of this study is to extend the guarantees provided by the secure transmission protocols such as Secure Sockets Layer (SSL) or Transport Layer Security (TLS) and apply them to the application layer. This paper proposes a comprehensive scheme that allows the unification of multiple security mechanisms, thereby removing the burden of authentication, mutual authentication, continuous authentication, and session management from the application development life-cycle. The proposed scheme will allow creation of high-level security mechanisms such as access control and group authentication on top of the extended security provisions. This scheme effectively eliminates the need for session cookies, session tokens and any similar technique currently in use. Hence reducing the attack surface and nullifying a vast group of attack vectors.

INDEX TERMS Authentication, multi-factor authentication, password-less authentication, application layer security, session management, cookies, tokens.

I. INTRODUCTION

The advent of technology and reliance on it has provided us with the comfort of on-demand availability of information systems on portable devices. Banking, healthcare, education etc. [1], [2], all have seen major improvements in remote availability. The increased availability and complexity, gives rise to the risk of an information system being compromised by malicious agents. Compromised information system can have severe implications in real world, ranging from identity theft to nuclear meltdowns [3]. Conventional security systems based on protocols such as SSL, Transport Layer Security (TLS) are considered secure. This is valid for data in motion and does not cater for data in use by the information systems. Hence the information system remains vulnerable to attacks which are not deterred by the security guarantees provided by protocols such as SSL and TLS. TLS does provide provable security during the initial handshake process which is based on hard problems such as integer factorization, discrete logarithm problem, or elliptic curve discrete logarithm problem. The underlying hard problem is dictated by the selected cipher suite. Transportation of the data on the other hand is usually symmetrically encrypted which is assumed to be secure [4]. Applications however rely on application layer sessions that are distinct from the transport layer sessions. Usually application session management mechanisms are developed by application developers who have limited to nonexistent knowledge of security. This has led to serious design flaws being present in widely used applications [5].

Security is considered a nonfunctional requirement and given low consideration during the software development life-cycle (SDLC) [6]. Standardization of session mechanisms on the application layer is insufficient, and there is a lack of readily available frameworks [7]. It is due to these issues that application development process remains highly prone to problems like logic errors, improper coding etc. [8]. The diverse nature of systems usually makes it very difficult to provide a unified security architecture that could be used not only to secure data in transit but also the data that is being processed at the application layer. Every application designer must implement their own security procedures on

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

the application layer which is usually prone to catastrophic failures [7].

An information system's integrity is dependent upon the authentication phase. Problems with authentication can either lead to lack of provisioning or to provisioning of resources to an adversary. Often systems are compromised due to problems at the authentication phase [9]. Once a user is authenticated there is a need to keep the user authenticated for the duration of interactions with the system without involving explicit re-authentication. This process is known as session management [10]. Session management is one of the most critical areas where the security of a system can be compromised e.g. web-based systems rely on cookies and tokens as a primary source of continual authentication. Mechanisms like cookies and tokens are susceptible to attacks and are a reason of enlargement of the attack surface. After the analysis of current network technologies; a lack of standardization for session management at application level is evident, whereas, there is extensive standardization at transport level [7]. An abstract discordance between these layers leads to the application layer being unable to benefit from the session management at the lower levels. Sessions if extended to the application layer will not only fetch better security for the system but also dramatically reduce the amount of effort that is currently being put into securing systems.

The registration phase in many currently deployed systems is considered a source of collecting authentication information. Whereas, this study emphasizes on the extension of the registration phase to serve as a key exchange or a key setup phase. Most systems leverage multi-factor authentication. The same can be used to facilitate key setup and key revocation. Such a setup will allow mutual authentication as compared to the server authentication in the traditional scenarios. Trust establishment between parties communicating over insecure channels has always proven to be a challenging problem, this scheme also works in environments where the traditional trust establishment infrastructure (PKI) is not suitable. Similarly the proposed scheme can be used to establish trust in situations where shifting burden of trust to a third party is not an option [11]. An example scenario is the use of Border Gateway Protocol (BGP) to acquire bogus TLS certificates [12] and the sale of counterfeit TLS certificates on the dark web [13]. The scheme can be implemented in a multiparty scenario where any number of participants can access resources in both individually identifiable, and a private mode where an individual's identity is concealed by the group. This paper looks at the feasibility of the proposed system to form the basis for an access control mechanism.

A. CONTRIBUTIONS

The main contributions are listed as follows:

- This study highlights the lack of a unified security architecture and discusses the issues that occur because of the limited cryptographic support TLS provides to the application layer.
- A novel mechanism is proposed which uses keys as an alternative to passwords and relies on cryptographically secure primitives for authentication, mutual authentication, continuous authentication, session management and other functionalities that are based on these mechanisms such as access control.
- The paper proposes an alternative methodology to the traditional Public Key Infrastructure (PKI) for mutual authentication in web-based applications.
- A client-server system is implemented that relies on the proposed scheme. The implementation is tested on an http server to achieve functionality similar to https without using PKI, tokens or cookies.

B. ORGANIZATION

The rest of the paper is organized as follows; in section II, currently used authentication techniques are discussed. Scheme primitives have been discussed in Section III. Section IV contains details of the proposed scheme. Section V details the implementation setup and results. Section VI concludes with a summary and discusses the future work.

II. AUTHENTICATION TECHNIQUES

The very first and the most important aspect of accessing a digital device or a service is the identification of a legitimate user authorized to access the digital device or service. Authentication [14] serves as the very basic countermeasure that ensures that only authorized users are granted access and unauthorized users are denied access. Numerous techniques and methodologies have been developed over the years that serve the purpose of authentication specifically. Distributed applications such as web applications are inherently stateless [15] that require retransmission of authentication data with every service request. Such a paradigm for continuous authentication [14] is disruptive for the seamless operation of a service and therefore requires an automated methodology for the retransmission of authentication credentials e.g. application sessions.

Authentication is one of the major challenges that is faced by any information system. Without having a foolproof authentication mechanism an information system is highly susceptible to a multitude of malicious intrusions. Some major authentication techniques are discussed below.

A. PASSWORD-BASED AUTHENTICATION

Password-based authentication serves as the primary authentication mechanism. Passwords are easy to deploy but they can also be termed as an aging authentication technique [16]. Written down passwords pose a significant security risk; therefore, sufficiently secure and usable passwords must be memorized. Memorizing random strings of alphanumeric data with a mix of special characters tends to get tedious as the size of passwords grow. Real world choices of passwords use a limited character set to keep the passwords manageable with regards to memorability [17]. By doing so one effectively reduces the total possible combinations available as passwords. Password choices also tend to skew towards the usage of meaningful words. These factors cumulatively reduce the entropy of passwords by a very huge factor. These weak passwords are easily susceptible to guessing, brute force and dictionary attacks [18].

The skewed nature of passwords has been addressed by Wang *et al.* in [19] by using 14 large scale datasets which includes 113.3 million real world passwords. Their work highlights the distribution of passwords generated by human users, along with providing a methodology to ascertain the security strength of a password dataset. They have used natural language processing techniques [20] to propose two models for the distribution of passwords namely ''PDF-Zipf'' and ''CDF-Zipf''. The researchers based on their findings proposed a new metric for the calculation of effective entropy of a usable password dataset. The metric depends upon the size of the dataset used and the number of successful guesses during an optimal attack [21], [22]. Their system was trained on English datasets, e.g. ''Flirtlife.de'' and ''Rockyou'' password datasets, however with the increasing support for Unicode [23] in online services the prevalence of passwords generated in other languages has increased. Our search for the word ''أُردُو'' in both ''Flirtlife.de'' and ''Rockyou'' returned empty. These findings suggest that their proposed models may not be generalizable to support all languages.

B. KEY-BASED AUTHENTICATION

Key-based authentication techniques [24], [25] belong to the category of authentication techniques based on something you have. Unlike passwords that require the authenticatee (an entity that is to be authenticated) to transmit the authentication data, key-based authentication schemes allow authentication based on the solution to a complex mathematical problem [26]. Such schemes allow the exchange of authentication information without transmitting the keys that are used as a replacement for passwords. It therefore provides higher security by not allowing the authentication information to be eavesdropped. Storing public keys on the server side in databases also has the added advantage of being immune to theft in case of server side breaches and database enumeration attacks [27], [28].

In [29] a novel authentication scheme has been devised using asymmetric cryptography based on generalized discrete logarithm problem and integer factorization problem. They have demonstrated their scheme to be highly efficient as it performs significantly better than self-certified schemes [30]. Key-based authentication schemes are usually computationally intensive because of the use of public key cryptography and therefore pose a challenge for low performance devices such as the ones used in IoT. Sciancalepore *et al.* present a novel scheme that allows low performance IoT devices to successfully achieve agreement with reduce overheads [31].

C. MULTI-FACTOR AUTHENTICATION

The issues faced while using traditional authentication mechanisms and the fact that authentication can be achieved based on something you know, something you are and something you have has led to the development of the multi-factor authentication paradigm [32]. Multi-factor schemes tend to incorporate two or more than two authentication factors. Alizai *et al.* [26] proposed a secure multi-factor device authentication scheme which uses digital signatures and device's capability to perform secure authentication. The multi-factor approach presented in their scheme uses device capability as a novel basis thereby protecting against attacks like man-in-the-middle and replay attacks.

Multi-factor authentication provides a higher security as compared to traditional authentication mechanisms [33]. However, the inclusion of a multi-factor authentication mechanism also increases the attack surface and introduces new attack vectors. Multi-factor authentication that relies on out of band communication channels can be exploited based on the vulnerabilities of the side channel [34]. For example, spoofing of authentication SMS and calls using a software-defined base station with a higher signal strength. These attack vectors can prove devastating for systems that put undue trust in the side channels integrity. Base station spoofing and a plethora of other vulnerabilities in the GSM network have been highlighted in the study [35].

D. BIOMETRIC AUTHENTICATION

Everyone has ''one of a kind'' attributes, some of which can be utilized to remarkably distinguish individuals. These qualities are extensively partitioned into physiological and behavioral biometrics. These biometric attributes incorporate iris, retina, fingerprint, palmprint, footprint, DNA, facial, voice, signatures or keyst-roke recognition respectively. Rathod *et al.* [36] have conducted a comprehensive study on the fingerprint biometric recognition systems. The study covers in detail false acceptance rate and false rejection rate. The paper also discusses limitations of each scheme that they analyze.

Multiple biometric [37] features instead of single biometric feature can also be used to authenticate a person more effectively and securely. In scenarios where multiple biometric features are used to authenticate a person, an attacker will have to create and misrepresent all the diverse sorts of biometric information. For example, acquiring a fingerprint and an iris image of adequate quality will be a difficult task for the attacker, thereby, making the attack difficult. Multi-biometrics framework is presented in [37] where different issues and tradeoffs are discussed in detail when designing such a multi-biometric framework.

E. HARDWARE AUTHENTICATION

Another perspective that can exceptionally authenticate an entity is the ownership of something. Things like smartcards [38], USB Security keys [39], RFID labels [40] can be utilized for authentication. Some of which may even utilize physical device attributes, for example, Physically Unclonable Functions (PUFs) [41]. PUFs are based on the distinctive properties of the equipment. These properties are incredibly hard to duplicate as they are based on a substantial number of factors including environmental noise and inherent material characteristics.

A lot has been done in the wake of standardizing the hardware-based authentication however it remains an area of active research. Some standard hardware-based security implementations include Hardware Security Modules (HSMs) [42]. HSMs are based on the idea of having a separate cryptographic co-processor which can provide cryptographic capabilities to other devices as an attached peripheral or over a network. HSMs do come with a hefty purchase price. Manufacturers of newer high end computational devices have started incorporating Trusted Platform Modules (TPM) [43] into their devices. TPMs resemble HSMs a lot in their functionality however they have a one to one relation with the device in which they are built in at the time of manufacture. Once a TPM is compromised it is not feasible to just replace the hardware and revocation of the keys also presents a major challenge [44].

III. SCHEME PRIMITIVES

This section highlights the scheme primitives that are helpful for understanding the scheme presented in the next section. This section also discusses the advancements in relative literature.

A. SESSION MANAGEMENT

After every successful authentication, there is usually a requirement for a procedure that allows subsequent requests to be authenticated without having to repeat the authentication phase. Session management techniques allow state-fullness and authentication without the overhead of re-authenticating. HTTP being stateless, achieves this stateful behavior by using cookies and tokens [45]. TLS achieves session management using IDs being sent as a part of the server "hello" message [4]. A technique for secure session management based on shared secret has been proposed in [10], which uses a simple incremental counter and HMAC for session management [46] but it has some performance and networking overhead.

Session management has been standardized for use in transport layer protocols like TLS, however the same cannot be said for application layer. Session management security issues have been ranked second in the Open Web Application Security Project (OWASP) top ten application security problems [9]. Session management mechanisms at the transport level do not provide enough session information to the application layer and stateless applications have intermittent transport layer sessions that span over single request response pair [15]. Every new request initiates a new session that is not related to the previous request. Most applications create their own independent session management. A partial goal of this study is to propagate the sessions established at transport layer to the application layer. Thereby effectively merging independent sessions at different layers of the network stack and providing a strong cryptographic basis for the resulting sessions. Ultimately leading to a strong standardization of the session management process and effectively reducing the application development effort that must be put into custom built session management.

B. COOKIES AND TOKENS

Cookies and tokens are the most widely employed continual authentication mechanisms used over the world wide web. HTTP and HTTPS both rely on cookies to provide stateful sessions. A forged cookie will result in the compromise of a system that has built its trust based on cookie's integrity. A wide array of techniques are available to the attacker for faking an application session that mainly include but not limited to session fixation [47], cookie theft [48] and token forgery [47]. A stolen cookie [49] can be used to set up a malicious session, although the attacker does not possess any information about the authentication information. The duration for which a session remains active needs to be optimized. Shorter durations tend to become a hurdle in the smooth working of the system. Similarly longer durations are severely vulnerable as they provide ample time for the attacker to conduct its malicious activity.

Stateless environments put the burden of implementing proper mechanisms on the system developers [47]. A stateless solution to the issues faced with cookies is tokens. Tokens are much like cookies however they are not persistent and do not require local storage. Tokens are built into the application logic e.g. secret strings appended to the URL or post data embedded into web pages. Just like cookies they are required to be transmitted along with every request however the token secret needs randomization after a few uses or else it can result in session fixation attacks [50]. Random number generation on such a massive scale, which can be used to effectively generate a token with enough entropy is usually a computationally costly operation. Invalidating a token requires a massive blacklist to be kept. This blacklist must be searched and accessed for every request being processed.

A comprehensive study was conducted on the viability and privacy concerns regarding cookies used by the top 100K Alexa websites [51]. Their analysis shows a very high prevalence of extremely insecure cookies. Zheng *et al.* [52] found cookie related vulnerabilities to be present in important sites including Google and Bank of America. They further analyzed the adverse effects on these vulnerabilities due to weak implementations in browsers that are widely being used.

C. TRANSPORT LAYER SECURITY MECHANISMS

Security of the data in motion is of major importance for any information system. One of the most used protocols for the security of data in motion is TLS. Besides security, the protocol is used to prevent eavesdropping, tampering and message forgery. Different cipher suites [4] can be used with TLS. Each cipher suite has its own underlying cryptographic primitives thereby provisioning different aspects and levels of security. TLS incorporates a variety of key agreement methodologies as well.

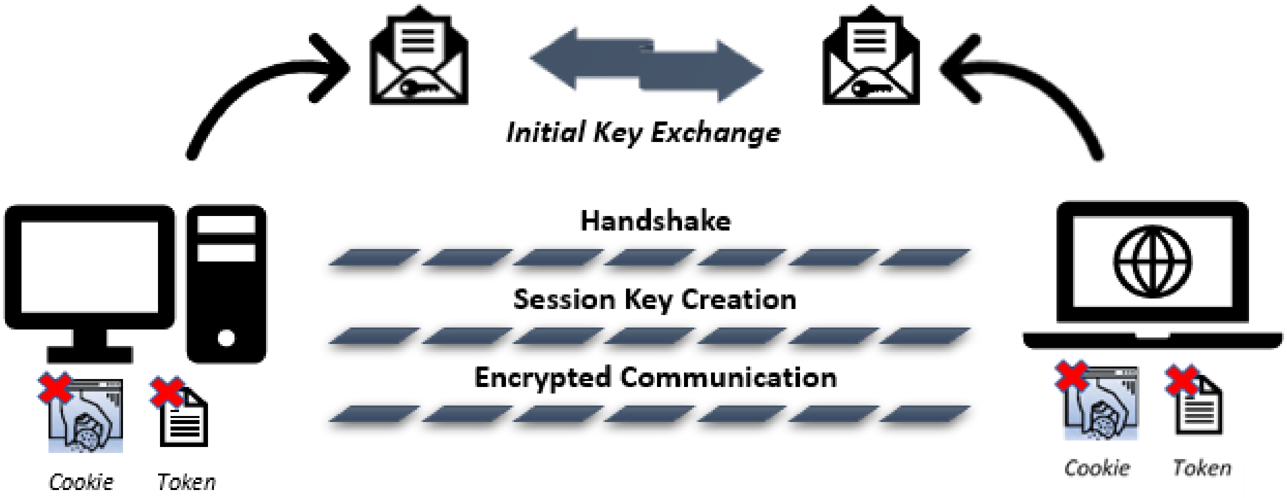


FIGURE 1. Proposed scheme.

TLS mainly focuses on the establishment of a secure communication channel [4]. Another study proposes an approach that is based on public key authentication for clients over networks such as the internet. They propose a TLS extension called TLS Origin Bound Certificates (TLS-OBC). This TLS extension allows clients of a system to establish strong authenticated channels with their servers. They use the technique to bind existing authentication tokens such as HTTP cookies to the authenticated channels. This scheme achieves a higher security for authentication using OBC. The scheme is typically feasible for existing world wide web infrastructure [53]. It is usually difficult to segregate TLS traffic based on the service it belongs to. Kim *et al.* [54] have proposed a novel approach to generate service signatures from the payload data of the TLS packets automatically. Their scheme achieves 90% efficiency of classifying TLS traffic according to the application/services they belong to.

IV. PROPOSED SCHEME

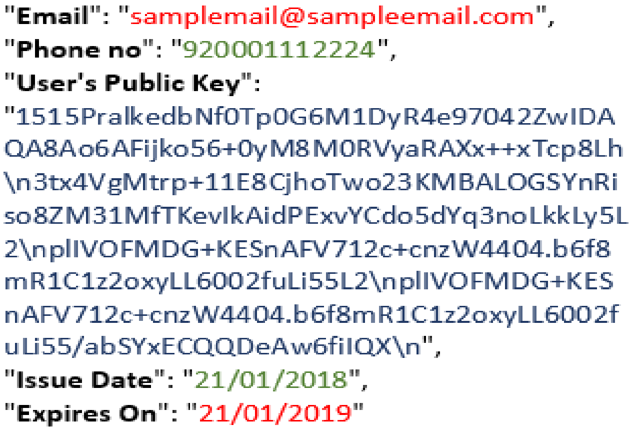
In the proposed scheme authentication, continuous authentication and session management are achieved at application layer by exploiting the fact that every service that requires user authentication has an associated registration phase. Figure 1 illustrates the public key exchange being carried out via side channels typically used for multi-factor authentication such as email etc. Once the key exchange has been carried out, a secure session can be established without using cookies or tokens at any time by following the steps shown in the figure 1. These steps ensure secrecy, aliveness, synchronization and immunity to Man-in-the-Middle attack. The steps are discussed in detail in the following sections. Table 1 explains the symbols used in the proposed scheme.

A. REGISTRATION PHASE

The registration phase has traditionally been used only as an account setup prerequisite, however in the proposed scheme, the registration phase is used not only in the traditional sense but also for the mutual authentication of both the service provider (Server) and the service requisitioner (Client). Most registration processes use side channel multi-factors such as email etc. This serves as the baseline for associating the public key to the digital identity presented at registration phase as illustrated in figure 2. Therefore, mutual authentication is very simple to establish given both the client and the server exchange their public keys during the registration phase.

TABLE 1. Table of notations.

Symbol	Notation
S	Server's Identity
C	Client's Identity
R.C	Client Random no.
R.S	Server Random no.
Enc_{Pub}	Encryption with Public Key
Dec_{PK}	Decryption with Private Key
$\|\|$	Concatenation



```
"Email": "samplemail@sampleemail.com",
"Phone no": "920001112224",
"User's Public Key":
"1515PralkedbNf0Tp0G6M1DyR4e97042ZwIDA
QA8Ao6AFijko56+0yM8M0RVyaRAXx++xTcp8Lh
\n3tx4VgMtrp+11E8CjhoTwo23KMBALOGSYnRi
so8ZM31MfTKevlkAidPExvYCdo5dYq3noLkkLy5L
2\nplIVOFMDG+KESnAFV712c+cnzW4404.b6f8
mR1C1z2oxyLL6002fuLi55L2\nplIVOFMDG+KES
nAFV712c+cnzW4404.b6f8mR1C1z2oxyLL6002f
uLi55/abSYxECQQDeAw6fiIQX\n",
"Issue Date": "21/01/2018",
"Expires On": "21/01/2019"
```

FIGURE 2. Sample certificate.

The client sends its public key to the service provider along with other registration information. This key does not require a certification authority to verify and sign it. The server will verify its ownership during the verification step using the multi-factor side channel e.g. an email containing verification information. Similarly, the server will send its own public key to the client via the same multi-factor side channel. The server's public key may or may not be certified by the certification authority, a better practice would be to have it certified.

B. AUTHENTICATION PHASE

Once both the server and the client have access to each other's public key information, it is possible for both server and client to mutually authenticate each other. Whenever the claimant sends a request to the server containing its identification information (ID tag), the server initiates a challenge-response scheme to authenticate the claimant. First, the server sends a random number R.S, server's identity S and a timestamp encrypted with the public key of the client as a challenge. The client decrypts the challenge and generates its random number R.C. Then client concatenates its random number R.C and client's identity C with the decrypted challenge, computes its hash and sends the hash along with the encrypted client's random number R.C with the public key of the server. Upon receiving the challenge, the server decrypts the received random number R.C, identity C, concatenates it with previously sent challenge and recomputes the hash of this newly created string. If the computed hash and the received hash are both equal at the server's side, this guarantees client authentication at server's side.

C. SESSION KEY ESTABLISHMENT

Now that both parties have each other's random number, symmetric keys are established based on the exchanged random numbers. They can easily create a symmetric session key which can be used in future requests without re-initiating the above challenge-response scheme again. To get a session key, both parties simply concatenate the random numbers and take hash of it using SHA 256. The resultant hash will be associated to the ID tag of the client on server side. The same hash will work as an AES 256-bit symmetric session key that will be used until the session expires. Once the symmetric session has been established, both client and server verify each incoming message by making sure that it has been encrypted using the same key that they computed during the key exchange. A message authentication code (MAC) [55] and client's ID tag is appended to every message that is transferred thereby enabling source authentication.

D. SCHEME HANDSHAKE

As illustrated in the figure 3, after sending an authentication request in step 1, the client must decrypt the challenge presented in step 2 to be capable of successfully completing the protocol negotiation. The same challenge also serves as a proof of client's identity. The data decrypted during step 3 is directly involved in the creation of the symmetric session key. Therefore, knowledge of the session key implies the knowledge of correct decryption in step 2 and hence proves the identity of the client. At server's side, a similar chain of events leads to the authentication of the server. As evident from the step 4 server is presented with an encrypted challenge, the correct decryption of which allows the server to obtain the information necessary for the creation of the session key. The knowledge of the correct symmetric session key serves as the proof of the server's identity. Hence both the client and the server are mutually authenticated.

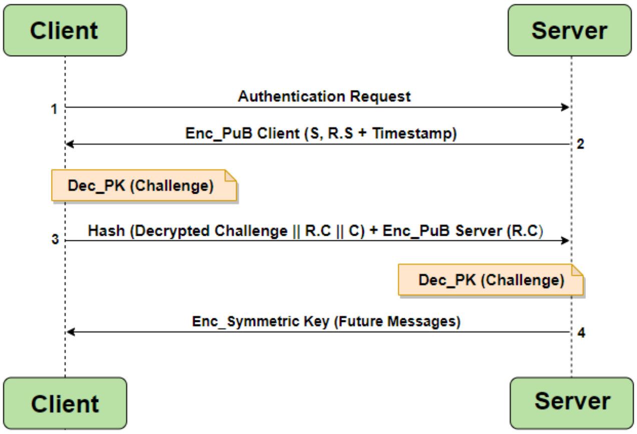

Client
Server
1
Authentication Request
Enc_PuB Client (S, R.S + Timestamp)
2
Dec_PK (Challenge)
3
Hash (Decrypted Challenge || R.C || C) + Enc_PuB Server (R.C)
Dec_PK (Challenge)
Enc_Symmetric Key (Future Messages)
4
Client
Server

FIGURE 3. Scheme handshake.

E. SECURITY PROVISIONS

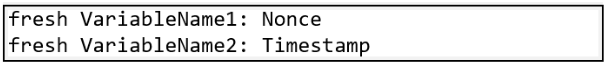
Authentication and continuous authentication are achieved by verifying each message. Session management in complex environments such as the world wide web depend heavily on stored artifacts such as the cookies and tokens. Verification success for a message implies that the message belongs to the session in question. Hence the need for cookies and tokens is eliminated. Higher level access control can be tied to the sessions to provision resources.

The proposed scheme mitigates cookie theft and token forgery attacks by eliminating the use of passwords, cookies or tokens for authentication, session management. The proposed scheme relies heavily on asymmetric key cryptography for authentication as well as session management.

a: DISCUSSION

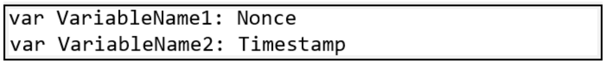
Following list enumerates the functionalities achieved through the scheme.

- This scheme provides mutual authentication between two or more parties.
- The scheme provides mutual authentication using a certificate exchange at registration phase.
- Sessions are created based on user authentication.
- Critical session information and cryptographic keys are derived for each session.
- The scheme provides continuous authentication based on the derived session key.



```
fresh VariableName1: Nonce
fresh VariableName2: Timestamp
```

FIGURE 4. Scyther script - 1.



```
var VariableName1: Nonce
var VariableName2: Timestamp
```

FIGURE 5. Scyther script - 2.

- All communication is symmetrically encrypted using TLS-PSK like cipher suites and integrity checks (MAC) are applied thus eliminating the need for cookies and tokens.
- Cipher suites that do not guarantee immunity to replay attacks requires an additional nonce encrypted with the same key, to be sent along with the encrypted data. This nonce is to be retained by the recipient during the active session.
- Ultimately resulting in secure sessions that are not vulnerable to attacks like cookie theft and token forgery.
- A session key remains usable for fix period thereby enforcing session timeouts.
- Session identifiers are mapped on to access control mechanisms which are in turn based on authentication information negotiated at registration phase.
- Keys derived from the authentication information and session information can be used to secure the communication.

V. SECURITY GOALS AND ANALYSIS

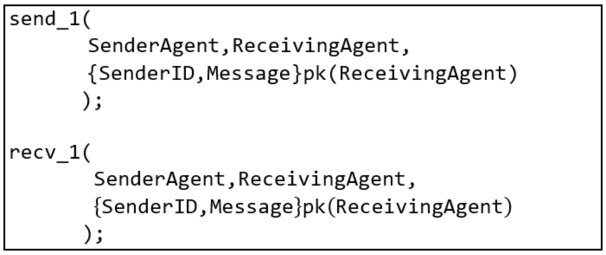
The proposed system has been analyzed using multiple analysis techniques. These techniques include simulating an adversary model using Scyther [56] based automated protocol verification tool and simulation of attacks such as man-in-the-middle attack leading to eavesdropping, packet mirroring, and replay of communication data. Communications were also analyzed for basic security provisions and the security goals were studied using Wireshark. This section briefly discusses the results extracted from the analysis.

A. SCYTHER-BASED ANALYSIS

The proposed scheme is analyzed with Scyther, which is an automated tool for protocol verification. Scyther script is used to test and validate the secrecy, aliveness, synchronization and resistance to man-in-the-middle properties of the proposed scheme. Scyther has its own scripting syntax which allows the simulation of protocols and their intruder models.

Scyther uses the keyword "fresh" when declaring a new variable. Figure 4 shows the nonce and timestamp creation.

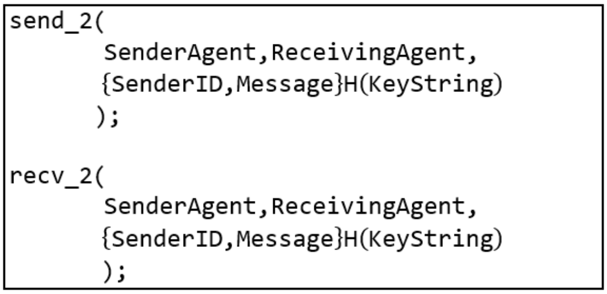
Keyword "var" is used when creating a new variable to store "fresh" type values that are received by another role. Figure 5 shows how new "var" type variables are created in Scyther.



```
send_1(
       SenderAgent,ReceivingAgent,
       {SenderID,Message}pk(ReceivingAgent)
      );

recv_1(
        SenderAgent,ReceivingAgent,
        {SenderID,Message}pk(ReceivingAgent)
       );
```

FIGURE 6. Scyther script - 3.



```
send_2(
        SenderAgent,ReceivingAgent,
        {SenderID,Message}H(KeyString)
       );

recv_2(
        SenderAgent,ReceivingAgent,
        {SenderID,Message}H(KeyString)
       );
```

FIGURE 7. Scyther script - 4.

The function send_1 and recv_1 simulate message transmission between the communicating agents. For every send function call in the sender agent there must be corresponding receive call in the receiving agent. Each of these calls are post-fixed with the same number indicating corresponding communication. Figure 6 shows a signed message being sent by the sender to the receiver.

Figure 7 shows a symmetrically encrypted message being sent by the sender to the receiver. The message is encrypted using the message digest of a string that corresponds to the key string that was generated during protocol handshake in our case.

A Network Threat Model [57] was simulated using Scyther script to analyze the security of the proposed scheme. The simulation was carried out under the following assumptions.

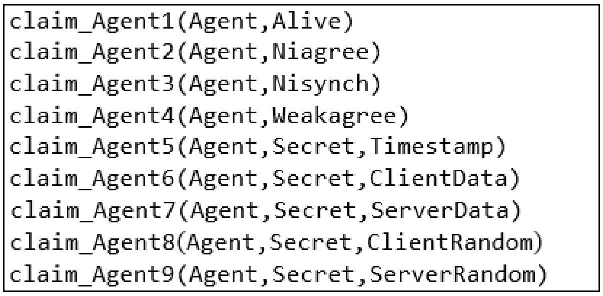
- The intruder is partially or fully in control of the network.
- The intruder can deflect, create and learn messages and is very powerful as defined by Dolev-Yao intruder model [57].

1) SECURITY VALIDATION

The verification of the scheme is given below according to the attributes of Scyther's tool.

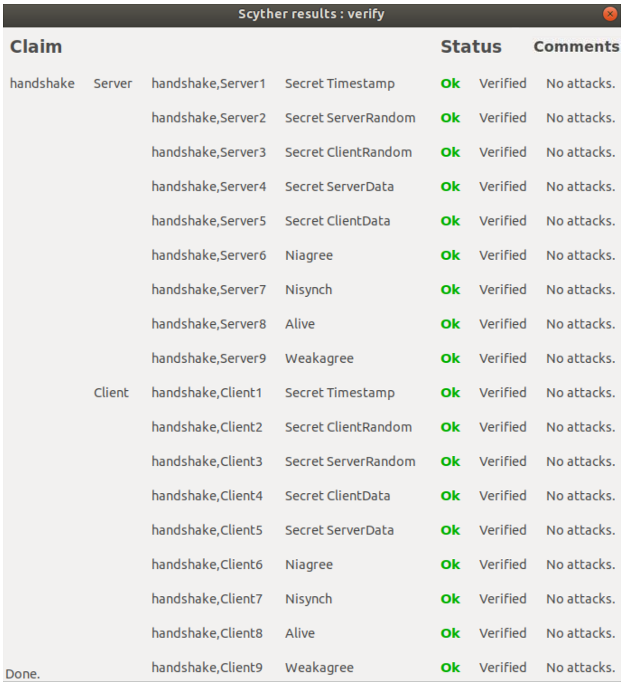
a: SECRECY

The first claim is that the scheme guarantees that users credentials will remain confidential. After analyzing the scheme, the credentials of both communicating parties are not revealed to any adversary when communicating over an untrusted network. The claims made for secrecy are shown in figure 8 as claim agent 5,6,7,8 and 9. Random numbers of



```
claim_Agent1(Agent,Alive)
claim_Agent2(Agent,Niagree)
claim_Agent3(Agent,Nisynch)
claim_Agent4(Agent,Weakagree)
claim_Agent5(Agent,Secret,Timestamp)
claim_Agent6(Agent,Secret,ClientData)
claim_Agent7(Agent,Secret,ServerData)
claim_Agent8(Agent,Secret,ClientRandom)
claim_Agent9(Agent,Secret,ServerRandom)
```

FIGURE 8. Scyther claim script.



Claim				Status		Comments
handshake	Server	handshake,Server1	Secret Timestamp	Ok	Verified	No attacks.
		handshake,Server2	Secret ServerRandom	Ok	Verified	No attacks.
		handshake,Server3	Secret ClientRandom	Ok	Verified	No attacks.
		handshake,Server4	Secret ServerData	Ok	Verified	No attacks.
		handshake,Server5	Secret ClientData	Ok	Verified	No attacks.
		handshake,Server6	Niagree	Ok	Verified	No attacks.
		handshake,Server7	Nisynch	Ok	Verified	No attacks.
		handshake,Server8	Alive	Ok	Verified	No attacks.
		handshake,Server9	Weakagree	Ok	Verified	No attacks.
	Client	handshake,Client1	Secret Timestamp	Ok	Verified	No attacks.
		handshake,Client2	Secret ClientRandom	Ok	Verified	No attacks.
		handshake,Client3	Secret ServerRandom	Ok	Verified	No attacks.
		handshake,Client4	Secret ClientData	Ok	Verified	No attacks.
		handshake,Client5	Secret ServerData	Ok	Verified	No attacks.
		handshake,Client6	Niagree	Ok	Verified	No attacks.
		handshake,Client7	Nisynch	Ok	Verified	No attacks.
		handshake,Client8	Alive	Ok	Verified	No attacks.
		handshake,Client9	Weakagree	Ok	Verified	No attacks.

FIGURE 9. Scyther result.

both server and client as well as the timestamp used in the negotiation remained secret as evident from figure 9.

b: ALIVENESS

The second claim is that the proposed technique achieves the aliveness property. This property ensures that the responding agent has executed an event in response to the communicating agent. It also ensures that the message exchange between the communicating parties has not been tampered, the messages are digitally signed and correctly time stamped. The claim made for aliveness is shown using the claim agent 1 in figure 8. The figure 9 depicts that the client and the server side achieve the aliveness property.

c: MAN-IN-THE-MIDDLE

Intercepting the communication during the handshake phase will require solving computational complex problems such as integer factorization, discrete logarithm problem or elliptic curve discrete logarithm problem. Similarly, the application data that is symmetrically encrypted will require brute force attacks. As no plain text data is available for the attacker to exploit so no attack tree was generated during any of the tests that were conducted in this research. Hence, it can be claimed that the proposed scheme is also resistant to man-in-the-middle attack. The claim for man-in-the-middle resistance is made using the claim agent 2 as shown in figure 8. Immunity against man-in-the-middle is achieved at both client and server side as evident from figure 9.

d: SYNCHRONIZATION

As described in man-in-the-middle section, an attacker will have no control over the system thereby limiting their capability for a replay attack. Mirrored/copied data might be resent to the recipient however the inclusion of nonce in sent data nullifies its effect. Hence the fourth claim that the proposed scheme is resistant against replay attacks is proved as it does not satisfy Non-injective Synchronization (Nisynch) property [58]. Nisynch property is used to ensure that the communication between sender and receiver is synced and sent by the sender. The claim made for synchronization is shown using claim agent 3 as shown in figure 8. The synchronization property has been achieved as evident from the results portrayed in figure 9.

2) SCYTHER RESULTS

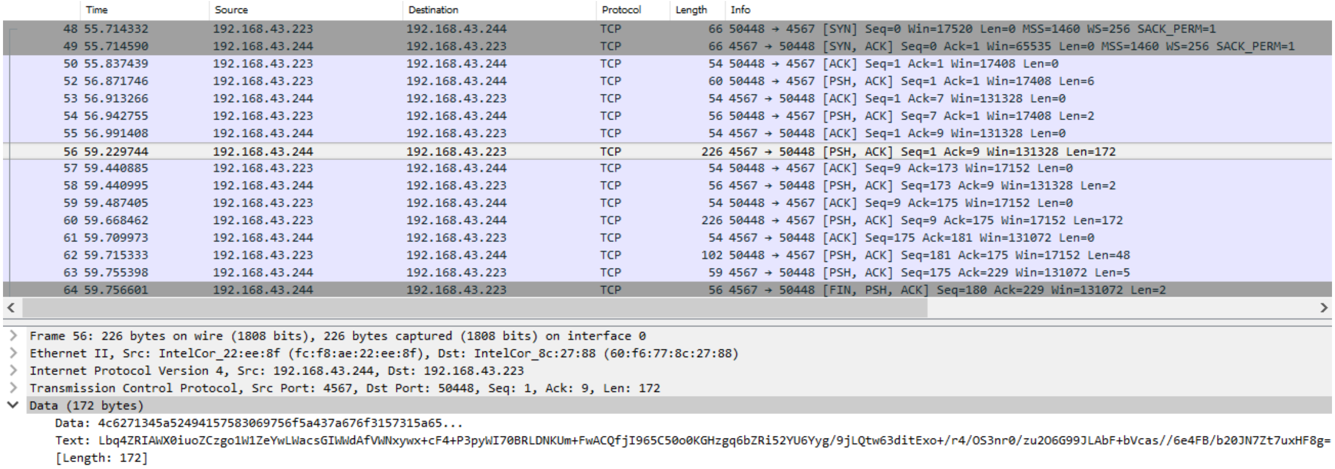
The Scyther script validated both communicating parties namely the server and the client separately. The figure 9 shows the results generated by Scyther. Scyther verified and validated the secrecy, aliveness, synchronization and resistance to man in the middle properties for the considered communicating parties.

Results of the Scyther analysis verify that ServerRandom, ServerData which were initialized as Nonce variables and ServerTimestamp which was initialized as a timestamp on the server's side remained secret. Similarly, the confidentiality of ClientRandom and ClientData, which were initialized as Nonce variables in the client role, is also verified.

B. WIRESHARK-BASED ANALYSIS

The proposed scheme has been tested separately in two environments. Firstly, a simple client server environment was setup. The keys for both client and server were exchanged over https using a simple registration form. Keys can also be exchanged manually by placing minimal certificates (containing only the corresponding public keys). An echo server was used to emulate the server side. The client-side application initiated a service request that was reciprocated by the server with an appropriate authentication challenge. Upon successful completion of the handshake stage both client and server agreed to a symmetric key.

The symmetric key was then used to instantiate a cipher object corresponding to AES-CBC [59] and AES-GCM [60] during different trials. Both client and server side were able to authenticate and keep the session running based on MAC verification (corresponding to the ID tag) *i.e.* a valid MAC



```
        Time          Source            Destination       Protocol  Length  Info
     48 55.714332     192.168.43.223    192.168.43.244    TCP         66 50448 → 4567 [SYN] Seq=0 Win=17520 Len=0 MSS=1460 WS=256 SACK_PERM=1
     49 55.714590     192.168.43.244    192.168.43.223    TCP         66 4567 → 50448 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
     50 55.837439     192.168.43.223    192.168.43.244    TCP         54 50448 → 4567 [ACK] Seq=1 Ack=1 Win=17408 Len=0
     52 56.871746     192.168.43.223    192.168.43.244    TCP         60 50448 → 4567 [PSH, ACK] Seq=1 Ack=1 Win=17408 Len=6
     53 56.913266     192.168.43.244    192.168.43.223    TCP         54 4567 → 50448 [ACK] Seq=1 Ack=7 Win=131328 Len=0
     54 56.942755     192.168.43.223    192.168.43.244    TCP         56 50448 → 4567 [PSH, ACK] Seq=7 Ack=1 Win=17408 Len=2
     55 56.991408     192.168.43.244    192.168.43.223    TCP         54 4567 → 50448 [ACK] Seq=1 Ack=9 Win=131328 Len=0
     56 59.229744     192.168.43.244    192.168.43.223    TCP        226 4567 → 50448 [PSH, ACK] Seq=1 Ack=9 Win=131328 Len=172
     57 59.440885     192.168.43.223    192.168.43.244    TCP         54 50448 → 4567 [ACK] Seq=9 Ack=173 Win=17152 Len=0
     58 59.440995     192.168.43.244    192.168.43.223    TCP         56 4567 → 50448 [PSH, ACK] Seq=173 Ack=9 Win=131328 Len=2
     59 59.487405     192.168.43.223    192.168.43.244    TCP         54 50448 → 4567 [ACK] Seq=9 Ack=175 Win=17152 Len=0
     60 59.668462     192.168.43.223    192.168.43.244    TCP        226 50448 → 4567 [PSH, ACK] Seq=9 Ack=175 Win=17152 Len=172
     61 59.709973     192.168.43.244    192.168.43.223    TCP         54 4567 → 50448 [ACK] Seq=175 Ack=181 Win=131072 Len=0
     62 59.715333     192.168.43.223    192.168.43.244    TCP        102 50448 → 4567 [PSH, ACK] Seq=181 Ack=175 Win=17152 Len=48
     63 59.755398     192.168.43.244    192.168.43.223    TCP         59 4567 → 50448 [PSH, ACK] Seq=175 Ack=229 Win=131072 Len=5
     64 59.756601     192.168.43.244    192.168.43.223    TCP         56 4567 → 50448 [FIN, PSH, ACK] Seq=180 Ack=229 Win=131072 Len=2

> Frame 56: 226 bytes on wire (1808 bits), 226 bytes captured (1808 bits) on interface 0
> Ethernet II, Src: IntelCor_22:ee:8f (fc:f8:ae:22:ee:8f), Dst: IntelCor_8c:27:88 (60:f6:77:8c:27:88)
> Internet Protocol Version 4, Src: 192.168.43.244, Dst: 192.168.43.223
> Transmission Control Protocol, Src Port: 4567, Dst Port: 50448, Seq: 1, Ack: 9, Len: 172
v Data (172 bytes)
    Data: 4c6271345a52494157583069756f5a437a676f3157315a65...
    Text: Lbq4ZRIAWX0iuoZCzgo1W1ZeYwLWacsGIWWdAfVWNxywx+cF4+P3pyWI70BRLDNKUm+FwACQfjI965C50o0KGHzgq6bZRi52YU6Yyg/9jLQtw63ditExo+/r4/OS3nr0/zu2O6G99JLAbF+bVcas//6e4FB/b20JN7Zt7uxHF8g=
    [Length: 172]
```

FIGURE 10. Traffic capture - Wireshark.

implies authenticity of the message origin. A simple access control was emulated using the session information. The communication was monitored using Wireshark (figure 10) during testing and the encryption was found to be comparable to that of TLS using AES-CBC or AES-GCM modes.

The same was tested using simple PHP server-side scripts and JavaScript running on client's browser. The results were very similar to the client server environment. Both the environments resulted in successful initialization of the symmetric key and encrypted communication. Additional testing for the http traffic was conducted using Burp Suite, however it was unable to produce any reconstruction of the data that was sent in the PHP setup except for the HTTP encapsulation of the HTML data that remained unencrypted as intended.

C. RESULT'S EXAMINATION

The testing conducted using Scyther and the implementation has shown positive results. All the presented functionality is successfully achieved. Certificates exchanged during registration are used to mutually authenticate both the communicating parties thus resulting in strong authentication. Authentication provides the baseline for the generation of session keys which are generated in the form of the shared secret. The shared secret can be used across multiple requests thus providing consistent sessions and continuous authentication. Forward secrecy is achieved by renegotiating the protocol for the generation of a new shared secret. As access control mechanisms are based on session information; this results in secure service provisioning.

The proposed system successfully allowed establishment of a secure communication channel between client and server. Mutual authentication was achieved in the absence of any of the traditional security mechanisms such as TLS. The initial key exchange was carried out using a secure TLS channel, however it is to be noted that the key exchange can leverage any of the channels commonly used by two factor authentications e.g. the server may issue keys to client via the usual out of band email or SMS channels. Successful session creation with valid reauthentication was established. Cookies and tokens were eliminated from the process. A basic access control criterion was found easy to implement using the session information that was setup. Environments lacking traditional security mechanisms can benefit from the scheme as it can be overlaid over existing systems.

VI. CONCLUSION

Authentication and session management are both pivotal to efficiently and effectively secure an information system. The security guarantees provided at lower layers of the network stack such as the transport layer provides a standardize and well tested basis for the issues faced by application layer session management implementations.

This paper presents an efficient and secure mutual authentication scheme that extends the registration phase's functionality to include the transfer of mutually authenticating information *i.e.* the public keys of both client and server. The scheme achieves authentication with a limited number of network transactions. A successful authentication results in the establishment of a shared secret that can be used in conjunction with a multitude of cipher suites. The scheme relies on identification data and message authentication codes appended to every subsequent message/request in-order to determine its authenticity.

The proposed scheme eliminates the attack vectors associated with cookies and tokens by entirely eliminating the need to use them as authentication information carriers. The proposed scheme, due to its lack of reliance in the PKI can prove to be very effective in scenarios that restricts PKI access.

The future work includes the study of hardware tokens as a source for authentication information. An extension of the current scheme using PGP and Blockchain for the initial key exchange is also underway.

REFERENCES

[1] L. Catarinucci, D. de Donno, L. Mainetti, L. Palano, L. Patrono, M. L. Stefanizzi, and L. Tarricone, "An IoT-aware architecture for smart healthcare systems," *IEEE Internet Things J.*, vol. 2, no. 6, pp. 515–526, Dec. 2015.

[2] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manuf. Lett.*, vol. 3, pp. 18–23, Jan. 2015.

[3] K. Kenney, "Cyber-terrorism in a post-stuxnet world," *Orbis*, vol. 59, no. 1, pp. 111–128, Jan. 2015.

[4] E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.3*, document RFC 8446, Aug. 2018.

[5] H.-Y. Shih, H.-L. Lu, C.-C. Yeh, H.-C. Hsiao, and S.-K. Huang, "A generic Web application testing and attack data generation method," in *Proc. Int. Conf. Secur. Intell. Comput. Big-data Services*. Springer, 2017, pp. 232–247.

[6] N. Nazir and M. K. Nazir, "American scientific research journal for engineering, technology, and sciences," *Amer. Sci. Res. J. Eng., Technol., Sci.*, vol. 42, no. 1, p. 166–187, Sep. 2018.

[7] K. Zhao and L. Ge, "A survey on the Internet of Things security," in *Proc. 9th Int. Conf. Comput. Intell. Secur.*, Dec. 2013, p. 663–667.

[8] A. One, "Smashing the stack for fun and profit," *Phrack*, vol. 7, no. 49, pp. 14–16, Nov. 1996.

[9] OWASP. (2010). *OWASP Top 10—2010: The Ten Most Critical Web Application Security Vulnerabilities*. Accessed: Apr. 1, 2019. [Online]. Available: https://owasptop10.googlecode.com/files/OWASP%5CnTop%5Cn10%5Cn-%5Cn2010.pdf

[10] P. De Ryck, L. Desmet, F. Piessens, and W. Joosen, "SecSess: Keeping your session tucked away in your browser," in *Proc. 30th Annu. ACM Symp. Appl. Comput.*, Apr. 2015, pp. 2171–2176.

[11] Y.-H. Lin, A. Studer, Y.-H. Chen, H.-C. Hsiao, L.-H. Kuo, J. M. McCune, K.-H. Wang, M. Krohn, A. Perrig, B.-Y. Yang, H.-M. Sun, P.-L. Lin, and J. Lee, "SPATE: Small-group PKI-less authenticated trust establishment," *IEEE Trans. Mobile Comput.*, vol. 9, no. 12, pp. 1666–1681, Dec. 2010.

[12] H. Birge-Lee, Y. Sun, A. Edmundson, J. Rexford, and P. Mittal, "Bamboozling certificate authorities with BGP," in *Proc. USENIX Secur. Symp. (SEC)*, 2018, p. 833–849.

[13] D. Maimon, Y. Wu, M. McGuire, N. Stubler, and Z. Qiu. *SSL/TLS Certificates and Their Prevalence on the Dark Web|Venafi*. Accessed: Jun. 19, 2019. [Online]. Available: https://www.venafi.com/sites/default/files/2019-02/Dark-Web-WP.pdf

[14] A. Al Abdulwahid, N. Clarke, I. Stengel, S. Furnell, and C. Reich, "Continuous and transparent multimodal authentication: Reviewing the state of the art," *Cluster Comput.*, vol. 19, no. 1, pp. 455–474, Mar. 2016.

[15] K. J. Kim, C. G. Kim, T. K. Whangbo, and K. Yoon, "A continuous playing scheme on RESTful Web service," *Cluster Comput.*, vol. 19, no. 1, pp. 379–387, Mar. 2016.

[16] K. Garrett, S. R. Talluri, and S. Roy, "On vulnerability analysis of several password authentication protocols," *Innov. Syst. Softw. Eng.*, vol. 11, no. 3, pp. 167–176, Sep. 2015.

[17] M. Lennartsson, "Evaluating the memorability of different password creation strategies: A systematic literature review," Univ. Skövde, Skövde, Sweden, Tech. Rep. IT610G, G2E, 22.5 HP, 2019.

[18] H.-J. Mun, S. Hong, and J. Shin, "A novel secure and efficient hash function with extra padding against rainbow table attacks," *Cluster Comput.*, vol. 21, no. 1, pp. 1161–1173, Mar. 2018.

[19] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2776–2791, Nov. 2017.

[20] J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science*, vol. 349, no. 6245, pp. 261–266, Jul. 2015.

[21] J. Bonneau, "The science of guessing: Analyzing an anonymized corpus of 70 million passwords," in *Proc. IEEE Symp. Secur. Privacy*, May 2012, p. 538–552.

[22] M. Dell' Amico, P. Michiardi, and Y. Roudier, "Password strength: An empirical analysis," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[23] *Unicode Consortium*. Accessed: Jun. 29, 2019. [Online]. Available: http://www.unicode.org/

[24] R. Amin and G. P. Biswas, "An improved RSA based user authentication and session key agreement protocol usable in TMIS," *J. Med. Syst.*, vol. 39, no. 8, p. 79, 2015.

[25] A. Witkovski, A. Santin, V. Abreu, and J. Marynowski, "An IdM and key-based authentication method for providing single sign-on in IoT," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, p. 1–6.

[26] Z. A. Alizai, N. F. Tareen, and I. Jadoon, "Improved IoT device authentication scheme using device capability and digital signatures," in *Proc. Int. Conf. Appl. Eng. Math. (ICAEM)*, Sep. 2018, pp. 1–5.

[27] L. Liu, J. Xu, C. Guo, K. Jiehui, S. Xu, and Z. Biao, "Exposing SQL injection vulnerability through penetration test based on finite state machine," in *Proc. 2nd IEEE Int. Conf. Comput. Commun. (ICCC)*, Oct. 2017, pp. 1171–1175.

[28] I. G. N. Mantra, M. Alaydrus, and H. M. Misni, "The Web security and vulnerability analysis model on indonesia higher education institution," in *Proc. Int. Conf. Inform. Comput. (ICIC)*, Oct. 2017, pp. 154–157.

[29] C. Meshram, C.-C. Lee, C.-T. Li, and C.-L. Chen, "A secure key authentication scheme for cryptosystems based on GDLP and IFP," *Soft Comput.*, vol. 21, no. 24, pp. 7285–7291, Dec. 2017.

[30] D. Li, H. Chen, C. Zhong, T. Li, and F. Wang, "A new self-certified signature scheme based on NTRUSing for smart mobile communications," *Wireless Pers. Commun.*, vol. 96, no. 3, pp. 4263–4278, Oct. 2017.

[31] S. Sciancalepore, G. Piro, G. Boggia, and G. Bianchi, "Public key authentication and key agreement in iot devices with minimal airtime consumption," *IEEE Embedded Syst. Lett.*, vol. 9, no. 1, p. 1–4, Mar. 2017.

[32] A. Ometov, S. Bezzateev, N. Mäkitalo, S. Andreev, T. Mikkonen, and Y. Koucheryavy, "Multi-factor authentication: A survey," *Cryptography*, vol. 2, no. 1, p. 1, Jan. 2018.

[33] D. Wang and P. Wang, "Two birds with one stone: Two-factor authentication with security beyond conventional bound," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 4, pp. 708–722, Jul./Aug. 2018.

[34] Z. Li, W. Wang, C. Wilson, J. Chen, C. Qian, T. Jung, L. Zhang, K. Liu, X. Li, and Y. Liu, "FBS-Radar: Uncovering fake base stations at scale in the wild," in *Proc. 24th Netw. Distrib. Syst. Secur. Symp. (NDSS)*, Mar. 2017, pp. 1–15.

[35] M. Pannu, R. Bird, B. Gill, and K. Patel, "Investigating vulnerabilities in GSM security," in *Proc. Int. Conf. Workshop Comput. Commun. (IEMCON)*, Oct. 2015, pp. 1–7.

[36] V. J. Rathod, N. C. Iyer, and S. M. Meena, "A survey on fingerprint biometric recognition system," in *Proc. Int. Conf. Green Comput. Internet Things (ICGCIoT)*, Oct. 2016, pp. 323–326.

[37] R. Gad, N. El-Fishawy, A. El-Sayed, and M. Zorkany, "Multi-biometric systems: A state of the art survey and research directions," *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 6, pp. 128–138, 2015.

[38] Q. Jiang, J. Ma, G. Li, and X. Li, "Improvement of robust smart-card-based password authentication scheme," *Int. J. Commun. Syst.*, vol. 28, no. 2, pp. 383–393, Jan. 2015.

[39] M. A. Wala'a and H. Abusaimeh, "Modified USB security token for user authentication," *Comput. Inf. Sci.*, vol. 8, no. 3, p. 51, Aug. 2015.

[40] A. X. Liu and L. A. Bailey, "PAP: A privacy and authentication protocol for passive RFID tags," *Comput. Commun.*, vol. 32, nos. 7–10, pp. 1194–1199, May 2009.

[41] P. Gope, J. Lee, and T. Q. S. Quek, "Lightweight and practical anonymous authentication protocol for RFID systems using physically unclonable functions," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2831–2843, Nov. 2018.

[42] D. Fox, "Hardware security module (HSM)," *Datenschutz und Datensicherheit-DuD*, vol. 33, no. 9, p. 564, Sep. 2009.

[43] T. Caddy, S. W. Smith, and A. Stavrou, "Trusted platform module," in *Encyclopedia of Cryptography and Security*. Boston, MA, USA: Springer, 2011, p. 1332–1335.

[44] *TPM Main Part 1 Design Principles Specification Version 1*, TCG, New York, NY, USA, 2011, p. 184.

[45] J. Jussila, "Http cookie weaknesses, attack methods and defense mechanisms: A systematic literature review," Fac. Inf. Technol., Jyväskylä Yliopisto, Jyväskylä, Finland, Tech. Rep. 61s, 2018.

[46] P. Kamal, "State of the art survey on session hijacking," *Global J. Comput. Sci. Technol.*, vol. 16, no. 1, pp. 39–49, Mar. 2016.

[47] S. Calzavara, A. Rabitti, and M. Bugliesi, "Dr cookie and Mr Token—Web session implementations and how to live with them," in *Proc. CEUR Workshop*, 2018, pp. 1–10.

[48] K. Lacroix, Y. L. Loo, and Y. B. Choi, "Cookies and sessions: A study of what they are, how they work and how they can be stolen," in *Proc. Int. Conf. Softw. Secur. Assurance (ICSSA)*, Jul. 2018, pp. 20–24.

[49] W.-B. Lee, H.-B. Chen, S.-S. Chang, and T.-H. Chen, "Secure and efficient protection for HTTP cookies with self-verification," *Int. J. Commun. Syst.*, vol. 32, no. 2, p. e3857, Jan. 2019.

[50] A. Amira, A. Ouadjaout, A. Derhab, and N. Badache, "Sound and static analysis of session fixation vulnerabilities in PHP Web applications," in *Proc. 7th ACM Conf. Data Appl. Secur. Privacy (CODASPY)*, Mar. 2017, pp. 139–141.

[51] A. Cahn, S. Alfeld, P. Barford, and S. Muthukrishnan, "An empirical study of Web cookies," in *Proc. 25th Int. Conf. World Wide Web (WWW)*, Apr. 2016, pp. 891–901.

[52] X. Zheng, J. Jiang, H. Canada, and U. C. Berkeley, "Cookies lack integrity: Real-world implications," in *Proc. 24th USENIX Secur. Symp. (USENIX Secur.)*, 2015, pp. 707–721.

[53] M. Dietz, A. Czeskis, and D. S. Wallach, "Origin-bound certificates: A fresh approach to strong client authentication for the Web," in *Proc. 21st USENIX Conf. Secur. Symp.*, 2012, pp. 317–331.

[54] S.-M. Kim, Y.-H. Goo, M.-S. Kim, S.-G. Choi, and M.-J. Choi, "A method for service identification of SSL/TLS encrypted traffic with the relation of session ID and server IP," in *Proc. 17th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Aug. 2015, pp. 487–490.

[55] S. Kelly and S. Frankel, *Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec*, document RFC 4868, May 2007.

[56] C. Cremers. (2016). *Scyther Tool*. Accessed: May 24, 2019. [Online]. Available: https://people.cispa.io/cas.cremers/scyther/

[57] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[58] C. J. F. Cremers, S. Mauw, and E. P. de Vink, "Injective synchronisation: An extension of the authentication hierarchy," *Theor. Comput. Sci.*, vol. 367, nos. 1–2, pp. 139–161, Nov. 2006.

[59] D. McGrew and D. Bailey, *AES-CCM Cipher Suites for Transport Layer Security (TLS)*, document RFC 6655, Jul-2012.

[60] J. Salowey, A. Choudhury, and D. McGrew, *AES Galois Counter Mode (GCM) Cipher Suites for (TLS)*, document RFC 5288, Aug. 2008, pp. 1–8.

ZAHOOR AHMED ALIZAI received the B.S. degree in computer science from Qurtuba University, D.I. Khan, in 2015. He is currently an M.S. Scholar with the School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Pakistan, under the supervision of Dr. H. Tahir. His research interests include network security, computer security, information security management, and cryptography.

HASAN TAHIR received the B.E. degree in software engineering from Bahria University, Islamabad, Pakistan, the M.S. degree in software engineering from the College of College of Electrical and Mechanical Engineering (E&ME), NUST, and the Ph.D. degree in information security from the University of Essex, U.K. He is currently an Assistant Professor with the School of Electrical Engineering and Computer Science (SEECS), NUST. He specializes in computer security and the IoT. He actively researches applications of cryptography in one to one and group settings. His primary area of research is the use of physically unclonable functions for securing group of devices. He teaches courses related to applied cryptography, cyber security, information security management, cloud computing security, software engineering, and software requirements analysis and design. He has served as a Committee Member in many renowned IEEE conferences. He was a recipient of the University of Essex Doctoral Scholarship Award.

MALIK HAMZA MURTAZA received the B.S. degree in software engineering from COMSATS University Islamabad at Wah, in 2017. He is currently an M.S. Scholar with the School of Electrical Engineering and Computer Science (SEECS), NUST, Pakistan, under the supervision of Dr. H. Tahir. He is also a Research Associate with the National Cybersecurity Auditing and Evaluation Lab (NCSAEL), NUST. His research interests include cryptography, network security, information security management, and computer security.

SHAHZAIB TAHIR received the B.E. degree in software engineering from Bahria University, Islamabad, Pakistan, in 2013, the M.S. degree in information security from NUST, Islamabad, Pakistan, in 2015, and the Ph.D. degree in information engineering from City, University of London, U.K., in January 2019. He is currently an Assistant Professor with the Department of Information Security, NUST, and also the Chief Technical Officer of the City Defend Limited, U.K. His research interests include applied cryptography and cloud security. He has been a TPC Member of many international IEEE conferences. He is an alumni of InnovateUK CyberASAP. He is a Reviewer of the IEEE TDSC, the IEEE Journal of Biomedical and Health Informatics, IEEE ICC, *Elsevier FGCS*, *Springer Cluster Computing*, *Springer Sadhana*, *Springer Science China Information Sciences*.

KLAUS MCDONALD-MAIER is currently a Professor with the School of Computer Science and Electronic Engineering, University of Essex, U.K., where he leads the Embedded and Intelligent Systems Research Group. He is actively involved in research in Systems-on-Chip (SoC) architectures and autonomous systems. Specifically, his research concentrates on novel computer and embedded systems techniques and architectures, embedded systems and SoC design, development support/debug and technology to increase performance, and security and reliability. He has also involved in hardware and software architectures offering advanced processing power for robotics, image processing and other real-time critical applications under limited power constraints and applied AI techniques for real world problems, robot control, and embedded systems. His research work has resulted in more than 150 scientific journal and conference publications as well as 12 pending patents. He has held five EPRSC grants with substantial contributions from industry and several EU grants. He is a Fellow of the IET and a member of the VDE.