An Efficient and Secure Key Exchange Protocol Based on Elliptic Curve and Security Models

Ahmad Abusukhon
Computer Science Dept. IT Faculty
Al-Zaytoonah University of Jordan
Amman, Jordan
ahmad.abusukhon@zuj.edu.jo

Zeyad Mohammad
Computer Science Dept. IT Faculty
Al-Zaytoonah University of Jordan
Amman, Jordan
z.dosooq@zuj.edu.jo

Ali Al-Thaher
IT Faculty
Al-Zaytoonah University of Jordan
Amman, Jordan
ali.t@zuj.edu.jo

***Abstract*— nowadays, the success of many online applications relays on keeping the data sent through the global network secure and far away from hackers. To carry out this task, the two communicating parties must exchange keys during their session. Some of the key exchange protocols are called key agreement protocols. The Elliptic Curve-Diffie Hellman (ECDH) is one of the most efficient algorithms for securing data. The ECDH is more efficient than other traditional techniques such as Rivest–Shamir–Adleman (RSA) in terms of key size, computation and network bandwidth. The Authenticated Key Agreement (AKA) protocol is used for establishing a common session key between the two communicating parties. The common session key is used for subsequent cryptography goals. Most of the key agreement protocols (e.g. Menezes-Qu–Vanstone (MQV) family) generate one key per session therefore increasing the opportunities for guessing the session key. In this paper, we focus on developing an enhanced multiple sessions key which is based on ECDH. We propose an efficient and secure AKA protocol which is based on the ideas of the hashed MQV (HMQV), the YAK protocol as a robust key agreement based on public key authentication and multiple session keys. The proposed protocol generates multiple common keys per a session, where the generated common key depends on the static and ephemeral keys. Furthermore, the proposed protocol overcomes the attacks on the HMQV and YAK protocols and provides desirable security properties as compared with the related works in this paper.**

Keywords*— *Asymmetric cryptography, Elliptic Curve Cryptography, Multiple session keys, Ephemeral keys, Hashing, YAK, HMQV.

I. INTRODUCTION

In general, there are two main types of data encryption namely symmetric key encryption and asymmetric key encryption. In symmetric key encryption the same key is used for both encryption and decryption process on both communication side. The work carried out by Abusukhon et al. [1][2][3][4][5][6] are examples on this type.

In asymmetric encryption, there are mainly two types of keys; the public key (which is known to every one) and the private key which is used for encryption and decryption process. In this scheme, a digital certificate is used for discovering the public key. A digital certificate may include information such as the name of the organization that issued the certificate, the public key of the use, and the e-mail address. Thus, in the client-server model, when the server and the client want to communicate they send a request to the third party (an organization) in order to send them back the certificates where the public key can be extracted from the certificate.

There are various protocols based on public key encryption such as: RSA, Diffie-Hellman key exchange (DH), Elliptic Curve Cryptography (ECC), Elliptic Curve Diffie-Hellman (ECDH), ELGamal encryption algorithm [7], YAK protocol, Menezes-Qu–Vanstone (MQV) protocol and the Hashed MQV (HMQV) protocol.

Abusukhon et al. [8] proposed a hybrid encryption algorithm based on Diffie-Hellman protocol and Text-to-Image Encryption algorithm called the DHTTIE. In DHTTIE protocol the plaintext is encrypted into an image (set of pixels). The encryption key is not sent through the channel, but it is built on both sides using the Diffie-Hellman key exchange protocol. In their protocol, the plaintext is encrypted using tow level of encryption namely the ciphertext plane (CIP) and the correction plane (COP).

MQV, HMQV, and YAK protocols are the focus of this research. We describe them briefly in section II.

The first step in setting up a secure communication between two parties is to establish a key [9]. Key Establishment (KE) is a protocol that allows two communicating parties to share a secret key. In general, KE is divided into two types namely Key Transport Protocol (KTP) and Key Agreement Protocol (KAP). In KTP, the key is generated on one side of the communicating parties (A) and then sent securely to the other party (B). However, in KAP multiple parties create a shared secret by contributing and combining information to obtain the result. In other words, two or more parties agree on a key to be used for conforming the communication privacy and authentication between them [10]. The first Key Exchange Protocol was proposed by Diffie-Hellman in 1976 [11]. This protocol lacks the verification between the two communicating parties and thus it is vulnerable against the man-in-the-middle attack. Many protocols are proposed to tackle this problem by proposing authentication based on several methods. One of these methods is relying on the public key infrastructure [12]. In general, there are four types of keys used in KAP namely, private static key agreement key, public static key agreement key, private ephemeral key agreement key, and the public ephemeral key agreement key. The KAP must hold the following security requirements; known key security, forward secrecy, key-compromise impersonation resilience, unknown key- share resilience and key control [12][13].

II. RELATED WORK

A. ECDH, MQV, HMQV and YAK Protocols

The ECDH protocol is a public key agreement protocol which allow two parties to communicate with each other to

978-1-5386-7942-5/19/$31.00 ©2019 IEEE

establish a shared secrete key for use in symmetric key algorithm [14][15]. ECDH is one of the most efficient algorithms for securing data. It is more efficient than other traditional techniques (such as RSA) in terms of key size, computation and network bandwidth. The Authenticated Key Agreement (AKA) protocol is used for establishing a common session key between the two communicating parties. The common session key is used for subsequent cryptography goals. Most of the key agreement protocols (e.g. MQV family) generate one key per session therefore increasing the opportunities for guessing the session key. The MQV is one of the agreement protocols in the Diffie-Hellman family [16]. It provides key authentication and forward secrecy by combining static and ephemeral key pairs. In addition, it has been shown that the MQV protocol is vulnerable to an unknown key-share attack [17].

In the unknown key attack, if a third party (E), gets the certificate of one of the communicating parties (A's certificate) and then E associates his/her name with the A's public key and replaces the A's certificate with his/her own certificate in a key agreement protocol then the other communicating party (B) believes that he is agreeing on the key with E. The MQV is defined over any finite commutative group G with a hard discrete logarithm problem (e.g. an elliptic curve group or multiplicative group).

The HMQV protocol is an enhanced MQV protocol [18] with focusing on provable security. Comparing MQV with HMQV, the major change is that the HMQV drops some mandated verification steps in MQV including the Proof of Possession (PoP) check during the CA registration and the prime-order validation check of the ephemeral public key [19].

In general, there are two types of authenticated two-party key agreement protocols namely, the Password Authenticated Key Exchange (PAKE) protocol and the Authenticated Key Exchange (AKE) protocol [18] (e.g. Public Key Authenticated Key Exchange, PK-AKE [20]). The YAK protocol is one of the PK-AKE protocols. The YAK protocol satisfies robust session key security under the Computational Diffie-Hellman (CDH) assumption in the random oracle model.

III. THE RESEARCH PROBLEM

Most of the recent protocols which are based on ECC are performing one key per session making it easier for hackers to guess the key (e.g. Menezes-Qu–Vanstone (MQV) family). To tackle this problem, we propose to generate multiple session keys for each individual session based on hashing techniques and a hybrid of other security protocols as we describe later in this paper.

IV. THE PROPOSED PROTOCOL

The main aim of this research is to tackle the problem mentioned in section III. As a solution to this problem, we propose an efficient and secure AKA protocol which is based on the ideas of the hashed MQV (HMQV), the YAK protocol as a robust key agreement based on public key authentication and the idea of producing multiple session keys per session [21]. The different between the proposed protocol and other protocols [21] is that the proposed protocol generates the session keys based on both the static and the ephemeral keys whereas the previous work uses only the public ephemeral keys.

It is expected that the proposed protocol produces multiple session keys per session (produce individual session key each one minute) and thus it weaken the attacks against the key when it is exchanged through the Internet. In addition, the proposed protocol must ensure the following; confidentiality, authentication and integrity as well. To facilitate understanding the proposed protocol we first describe the main notations used in our proposed protocol as described in Table I.

TABLE I. NOTATIONS USED IN THE PROPOSED PROTOCOL

Symbol	Definition
C	Client
S	Server
G	Construct a new elliptic curve point with the given (x, y) coordinates
G	The generating point of ECC large Prime order in $\#E(F_q) = n$
$\langle G \rangle$	A cyclic subgroup of P
$\langle G^* \rangle$	The set of nonidentity elements ⟨P⟩
$\langle G_\infty \rangle$	The identity point in $E(F_q)$
c_1, s_1	Static private keys of C and S, where $c_1, s_1 \in_R [1, n-1]$
C_1, S_1	Static public keys of C and S, respectively.
$r_{1_C}, r_{2_C}, r_{3_C}, r_{1_S}, r_{2_S}, r_{3_S}$	Ephemeral private keys of C and S, respectively.
$R_{1_C}, R_{2_C}, R_{3_C}, R_{1_S}, R_{2_S}, R_{3_S}$	Ephemeral public keys of C and S, respectively.
H, H_1, H_2	$\{0,1\}^* \rightarrow F_{\vert q \vert}$: A collision of resistant hash functions which the output length is |q| bits.
H_3	$\{0,1\}^* \rightarrow \{0,1\}^\lambda$: Hash functions of modeled random oracle, where λ is a security parameter.
ZKP	Zero Knowledge Prove
$createsessionKey_{1-9}$	Create nine various session keys. $(createEphemeralKey1, .., createEphemeralKey9$
$\oplus$	XOR operation.
||	Concatenation
mod	modules

The proposed protocol is a two-pass protocol that generates nine keys per session and the security for a generated key depends on static and ephemeral keys thus thwart the imperfect random generation and side channel attacks. Furthermore, the proposed protocol uses a hash function to derive each session key that thwarts key-replication attack and validates the public key that thwarts the small subgroup attacks. Moreover, the proposed protocol uses Digital Signature Algorithm (DSA) as in the HMQV with a slightly modification in order to withstand against key compromise impersonation attacks, and it uses a Zero-Knowledge proof to verify the ephemeral private keys of its partner.

It is expected that the proposed protocol will overcome various type of attacks on the original protocols (HMQV and YAK) and provides desirable security properties. Next, we describe the proposed protocol.

Firstly, the public static keys of the client and server should be registered in the Certificate Authority (CA); thereafter they will get their certificates that bind their public static keys with their identities in order to use them in the proposed protocol for generating multiple keys per session. The client selects a static private key such that $c_C \in_R [1, n-1]$ and calculates $C_1 = c_1 G$. Similarly, the server selects a static private key such that $s_S \in_R [1, n-1]$ and calculates $S_1 = s_1 G$. The proposed protocol performs the following steps to generate nine keys per session as described in Figure 1(a, b, c, d and e):

Step 1.1 The client initially (as an initiator) selects three random numbers $r_{1C}, r_{2C}, r_{3C} \in_R [1, n-1]$ and calculates the ephemeral public keys as follows:

$$R_{1_C} = r_{1_C} G \tag{1}$$

$$R_{2_C} = r_{2_C} G \tag{2}$$

$$R_{3_C} = r_{3_C} G \tag{3}$$

Step 1.2 The client selects a random number:

$v_C \in_R [1, n-1]$

and then calculates V_C , the hash function h_C and Z_C as follows:

$$V_C = v_C \, G \tag{4}$$

$$h_C = H_1(G \parallel V_C \parallel R_{1_C} \parallel R_{2_C} \parallel R_{3_C}) \tag{5}$$

$$Z_C = v_C - (r_{1_C} + r_{2_C} + r_{3_C}) h_C \ mod \ n \tag{6}$$

then it sends $R_{1C}, R_{2C}, R_{3C}, V_C, Z_C$ to the Server.

Step 2.1 Upon receiving the client message, the server initially (as a responder) selects three numbers $r_{1S}, r_{2S}, r_{3S} \in_R [1, n-1]$ and calculates the ephemeral public keys as follows:

$$R_{1_S} = r_{1_S} G \tag{7}$$

$$R_{2_S} = r_{2_S} G \tag{8}$$

$$R_{3_S} = r_{3_S} G \tag{9}$$

Step 2.2 The server selects a random number $v_S \in_R [1, n-1]$ and calculates V_S, the hash function h_S and Z_S as follows:

$$V_S = v_S \, G \tag{10}$$

$$h_S = H_1(G \parallel V_S \parallel R_{1_S} \parallel R_{2_S} \parallel R_{3_S}) \tag{11}$$

$$Z_S = v_S - (r_{1_S} + r_{2_S} + r_{3_S}) h_S \ mod \ n \tag{12}$$

and Verifies the term $V_C \overset{?}{=} Z_C \, G + h_C(R_{1_C} + R_{2_C} + R_{3_C})$ if it holds then sends $R_{1S}, R_{2S}, R_{3S}, V_S, Z_S$ to the client and computes nine keys as shows in the Figure 1(a, b, c, d and e).

Step 3 Upon receiving the server message, the client verifies the $V_S \overset{?}{=} Z_S \, G + h_S(R_{1_S} + R_{2_S} + R_{3_S})$ if it holds then the client computes nine keys as described in Figure 1(a, b, c, d and e).

V. Security Analysis of the Proposed Protocol

In this section, we demonstrate the security analysis of the proposed protocol against various types of attacks.

A. *Known-Key Security*

A protocol execution should result in a unique common session key. If this key is revealed, it should have no impact on either a passive attack to compromise future session keys or an impersonation by an active attack in the future [22].

Suppose that all the common session keys ($k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9$) are disclosed to an adversary. The adversary cannot obtain any static secret key from the participating parties since the proposed protocol uses the hash function for deriving the session key, where the hash function is modeled a random oracle problem thus thwarting this attack on the proposed protocol.

B. *Replay*

The assumption in Table I, that is, the prime order n of point G is large such that the probability of a party selecting the same ephemeral private key in two different sessions is negligible. The σ in the proposed protocol depends on static secret key and ephemeral secret key of a party; thus, the adversary is unable to compute the session key even though in the case of the ephemeral private key is disclosed to the adversary.

C. *Forgery*

Assume that an adversary wants to impersonate a client to establish the common session keys with a server. The adversary forges the previously intercepted message and sends it to the server, the message of an adversary cannot pass the verification steps in the proposed protocol in the sense that we have in our proposed protocol the Zero Knowledge Prove (ZKP, ephemeral secret keys) mechanism for verification and thus thwarting this attack in the proposed protocol.

D. *Man-in-the-Middle Attack*

The adversary cannot lunch man-in-the-middle attack since the adversary cannot manipulate a message in order to cancel the secret terms from

$$\sigma_1 = (d.c_1 + r_{1_r})(e.S_1 + R_{1_S}) \tag{13}$$

where

$$d = H_2(R_{1C}, R_{1S}, C, S) \tag{14}$$

and

$$e = H_2(R_{1S}, R_{1C}, C, S) \tag{15}$$

Therefore, the proposed protocol is able to prevent these forms of attacks.

Client		Server
$r_{1C}, r_{2C}, r_{3C}, v_C \in_R [1, n-1]$		
$R_{1_C} = r_{1_C} G$		
$R_{2_C} = r_{2_C} G$		
$R_{3_C} = r_{3_C} G$		
$V_C = v_C G$		
$h_C = H_1(G \parallel V_C \parallel R_{1_C} \parallel R_{2_C} \parallel R_{3_C})$		
$Z_C = v_C - (r_{1_C} + r_{2_C} + r_{3_C}) h_C \bmod n$		
	Sends $R_{1C}, R_{2C}, R_{3C}, V_C, Z_C$ $\longrightarrow$	
		$h_C = H_1(G \parallel V_C \parallel R_{1_C} \parallel R_{2_C} \parallel R_{3_C})$
		$V_C \overset{?}{=} Z_C G + h_C(R_{1_C} + R_{2_C} + R_{3_C})$
		$r_{1S}, r_{2S}, r_{3S}, v_S \in_R [1, n-1]$
		$R_{1_S} = r_{1_S} G$
		$R_{2_S} = r_{2_S} G$
		$R_{3_S} = r_{3_S} G$
		$V_S = v_S G$
		$h_S = H_1(G \parallel V_S \parallel R_{1_S} \parallel R_{2_S} \parallel R_{3_S})$
		$Z_S = v_S - (r_{1_S} + r_{2_S} + r_{3_S}) h_S \bmod n$
	$\longleftarrow$ Send $R_{1S}, R_{2S}, R_{3S}, V_S, Z_S$	
$h_S = H_1(G \parallel V_S \parallel R_{1_S} \parallel R_{2_S} \parallel R_{3_S})$		
$V_S \overset{?}{=} Z_S G + h_S(R_{1_S} + R_{2_S} + R_{3_S})$		

Fig. 1 (a) The nine ephemeral keys

client	server
$createEphemeralKey1$	$createEphemeralKey1$
$d = H_2(R_{1C}, R_{1S}, C, S)$	$d = H_2(R_{1C}, R_{1S}, C, S)$
$e = H_2(R_{1S}, R_{1C}, C, S)$	$e = H_2(R_{1S}, R_{1C}, C, S)$
$\sigma_1 = (d.c_1 + r_{1_C})(e.S_1 + R_{1_S})$	$\sigma_1 = (e.s_1 + r_{1_S})(d.C_1 + R_{1_C})$
$k_1 = H_3(\sigma_1, R_{1C}, R_{1S}, C, S)$	$k_1 = H_3(\sigma_1, R_{1C}, R_{1S}, C, S)$
$createEphemeralKey2$	$createEphemeralKey2$
$d = H_2(R_{1C}, R_{2S}, C, S)$	$d = H_2(R_{1C}, R_{2S}, C, S)$
$e = H_2(R_{2S}, R_{1C}, C, S)$	$e = H_2(R_{2S}, R_{1C}, C, S)$
$\sigma_2 = (d.c_1 + r_{1_C})(e.S_1 + R_{2_S})$	$\sigma_2 = (e.s_1 + r_{2_S})(d.C_1 + R_{1_C})$
$k_2 = H_3(\sigma_2, R_{1C}, R_{2S}, C, S)$	$k_2 = H_3(\sigma_2, R_{1C}, R_{2S}, C, S)$

Fig. 1 (b) The nine ephemeral keys

E. *Perfect Forward Secrecy*

This property of key agreement protocols ensures that compromise of long-term keys does not compromise past session keys. It protects past sessions against future compromises of secret keys. Since the session keys of the proposed protocol are dependent upon random numbers r_C and r_S which are intractable to the adversaries for computing the term $r_C r_S G$ from the transmitted messages during the protocol run. This means that the proposed protocol is able to prevent this attack.

client	server
$createEphemeralKey3$	$createEphemeralKey3$
$d = H_2(R_{1C}, R_{3S}, C, S)$	$d = H_2(R_{1C}, R_{3S}, C, S)$
$e = H_2(R_{3S}, R_{1C}, C, S)$	$e = H_2(R_{3S}, R_{1C}, C, S)$
$\sigma_3 = (d.c_1 + r_{1_C})(e.S_1 + R_{3_S})$	$\sigma_3 = (e.s_1 + r_{3_S})(d.C_1 + R_{1_C})$
$k_3 = H_3(\sigma_3, R_{1C}, R_{3S}, C, S)$	$k_3 = H_3(\sigma_3, R_{1C}, R_{3S}, C, S)$
$createEphemeralKey4$	$createEphemeralKey4$
$d = H_2(R_{2C}, R_{1S}, C, S)$	$d = H_2(R_{2C}, R_{1S}, C, S)$
$e = H_2(R_{1S}, R_{2C}, C, S)$	$e = H_2(R_{1S}, R_{2C}, C, S)$
$\sigma_4 = (d.c_1 + r_{2_C})(e.S_1 + R_{1_S})$	$\sigma_4 = (e.s_1 + r_{1_S})(d.C_1 + R_{2_C})$
$k_4 = H_3(\sigma_4, R_{2C}, R_{1S}, C, S)$	$k_4 = H_3(\sigma_4, R_{2C}, R_{1S}, C, S)$

Fig. 1 (c) The nine ephemeral keys

F. *Key Compromise Impersonation*

If either client's static secret key and server's ephemeral secret key or client's ephemeral private key is compromised at one time, an adversary is able to impersonate the client. But this should not enable it to impersonate other parties of the client [22].

Assume that the static key of the client c_1 and the ephemeral private key of the server r_s are compromised at the same time, an adversary is not able to use them to impersonate the client as the other party since the adversary cannot compute the term $r_c s_1 G$ from the transmitted messages and the manipulated messages, and it cannot cancel the secret terms in σ by substituting messages during the protocol run. On the other hand, if the ephemeral private key r_C of the client is compromised by an adversary, the adversary is not able to compute the shared static key between two-party participants which is included in the session key equation.

G. Unknown Key Share

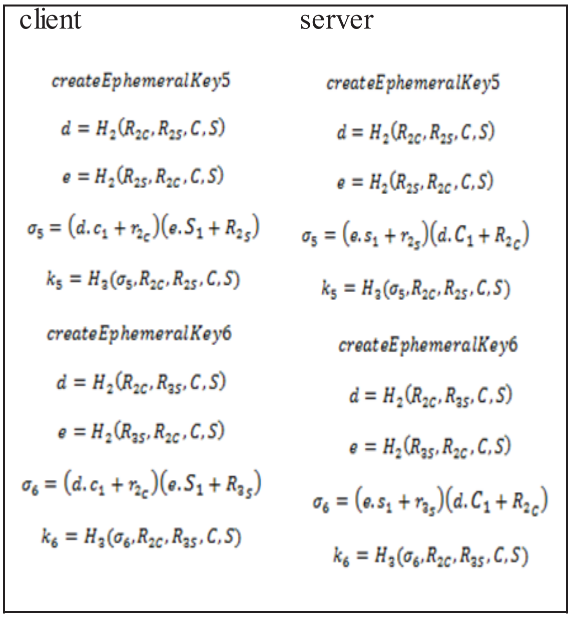
If the client wants to create a common secret key with the server, it should not be possible that the server is tricked into sharing a key with entity such as C [22]. Since the proposed protocol uses a hash function to derive the common session key that includes the identities of the two participating parties and the terms of d and e are hashed values, which depend on the identities of the two-party participants; therefore thwarting this attack.

H. Key Control

Both of the client and server should not be able to force the common session key to a value of their choice [21] since the session key equation:

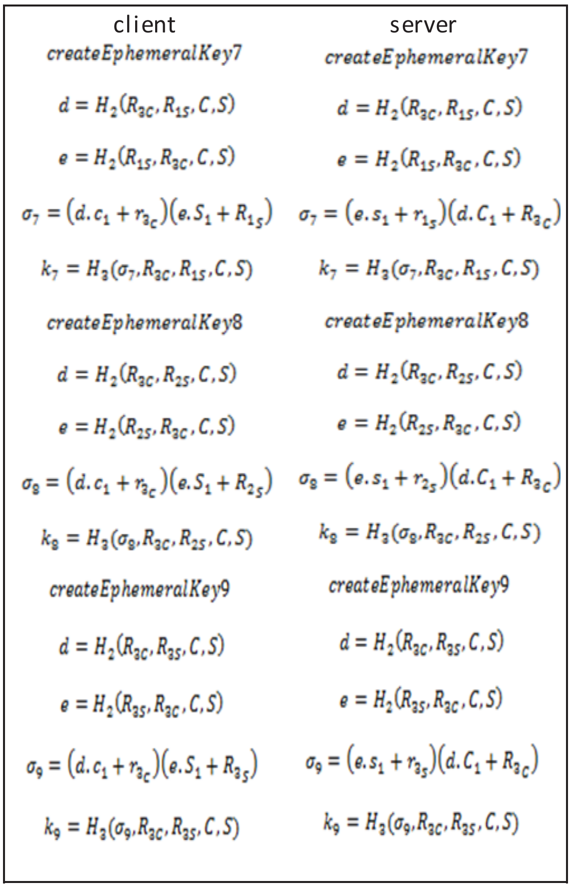
$$\sigma_1 = (d.c_1 + r_{1_c})(e.s_1 + r_{1_s})G \tag{16}$$

in the proposed protocol includes both ephemeral private keys from the two-party participants, which are selected randomly by the client and server.



client	server
$createEphemeralKey5$	$createEphemeralKey5$
$d = H_2(R_{2C}, R_{2S}, C, S)$	$d = H_2(R_{2C}, R_{2S}, C, S)$
$e = H_2(R_{2S}, R_{2C}, C, S)$	$e = H_2(R_{2S}, R_{2C}, C, S)$
$\sigma_5 = (d.c_1 + r_{2_c})(e.S_1 + R_{2_S})$	$\sigma_5 = (e.s_1 + r_{2_s})(d.C_1 + R_{2_C})$
$k_5 = H_3(\sigma_5, R_{2C}, R_{2S}, C, S)$	$k_5 = H_3(\sigma_5, R_{2C}, R_{2S}, C, S)$
$createEphemeralKey6$	$createEphemeralKey6$
$d = H_2(R_{2C}, R_{3S}, C, S)$	$d = H_2(R_{2C}, R_{3S}, C, S)$
$e = H_2(R_{3S}, R_{2C}, C, S)$	$e = H_2(R_{3S}, R_{2C}, C, S)$
$\sigma_6 = (d.c_1 + r_{2_c})(e.S_1 + R_{3_S})$	$\sigma_6 = (e.s_1 + r_{3_s})(d.C_1 + R_{2_C})$
$k_6 = H_3(\sigma_6, R_{2C}, R_{3S}, C, S)$	$k_6 = H_3(\sigma_6, R_{2C}, R_{3S}, C, S)$

Fig. 1 (d) The nine ephemeral keys



client	server
$createEphemeralKey7$	$createEphemeralKey7$
$d = H_2(R_{3C}, R_{1S}, C, S)$	$d = H_2(R_{3C}, R_{1S}, C, S)$
$e = H_2(R_{1S}, R_{3C}, C, S)$	$e = H_2(R_{1S}, R_{3C}, C, S)$
$\sigma_7 = (d.c_1 + r_{3_c})(e.S_1 + R_{1_S})$	$\sigma_7 = (e.s_1 + r_{1_s})(d.C_1 + R_{3_C})$
$k_7 = H_3(\sigma_7, R_{3C}, R_{1S}, C, S)$	$k_7 = H_3(\sigma_7, R_{3C}, R_{1S}, C, S)$
$createEphemeralKey8$	$createEphemeralKey8$
$d = H_2(R_{3C}, R_{2S}, C, S)$	$d = H_2(R_{3C}, R_{2S}, C, S)$
$e = H_2(R_{2S}, R_{3C}, C, S)$	$e = H_2(R_{2S}, R_{3C}, C, S)$
$\sigma_8 = (d.c_1 + r_{3_c})(e.S_1 + R_{2_S})$	$\sigma_8 = (e.s_1 + r_{2_s})(d.C_1 + R_{3_C})$
$k_8 = H_3(\sigma_8, R_{3C}, R_{2S}, C, S)$	$k_8 = H_3(\sigma_8, R_{3C}, R_{2S}, C, S)$
$createEphemeralKey9$	$createEphemeralKey9$
$d = H_2(R_{3C}, R_{3S}, C, S)$	$d = H_2(R_{3C}, R_{3S}, C, S)$
$e = H_2(R_{3S}, R_{3C}, C, S)$	$e = H_2(R_{3S}, R_{3C}, C, S)$
$\sigma_9 = (d.c_1 + r_{3_c})(e.S_1 + R_{3_S})$	$\sigma_9 = (e.s_1 + r_{3_s})(d.C_1 + R_{3_C})$
$k_9 = H_3(\sigma_9, R_{3C}, R_{3S}, C, S)$	$k_9 = H_3(\sigma_9, R_{3C}, R_{3S}, C, S)$

Fig. 1 (e) The nine ephemeral keys

CONCLUSION AND FUTURE WORK

In this research, we have developed a multiple session keys protocol that produces multiple session keys in one session. The security for a generated key depends on static and ephemeral keys thus thwart the imperfect random generation and side channel attacks. The proposed protocol differs from other protocols (YAK and the MQV family) in terms of the number of session keys produced in one session and thus making it hard for hackers to guess the session keys. In addition, the proposed protocol generates the session keys based on both the static and the ephemeral keys whereas the previous work uses only the public ephemeral keys.

To make it difficult for hackers to guess the session keys, we configured our system such that the session key is changed each one minute. However, this increases the number of hashing as well as the number of Point Multiplication (PM). In future, we attend to evaluate the security of the proposed system using standard cryptanalysis. The work carried out in [22][23] are examples on cryptanalysis. The proposed protocol can be applied to various aspects such as military, health, business, data mining [24][25][26][27], and software engineering [28][29].

ACKNOWLEDGMENT

We would like to express our sincere gratitude to Al-Zaytoonah University of Jordan for their financial support and encouragements.

REFERENCES

[1] A. Abusukhon, M. Talib, and I. Ottoum, "Secure Network Communication Based on Text-to-Image Encryption" International Journal of Cyber-Security and Digital Forensics (IJCSDF) The Society of Digital Information and Wireless Communications (SDIWC) vol.1, pp. 263-271, 2012.

[2] A. Abusukhon, M. Talib, and M. Nabulsi, "Analyzing the Efficiency of Text-to-Image Encryption Algorithm" International Journal of Advanced Computer Science and Applications (IJACSA), vol.3, pp 35-38, 2012.

[3] A. Abusukhon and M. Talib "Distributed Text-to-Image Encryption Algorithm" International Journal of Computer applications (IJCA). vol. 106, pp 1-5. 2014.

[4] A. Abusukhon "Block Cipher Encryption for Text-to-Image Algorithm" International Journal of Computer Engineering & Technology (IJCET).vol.4, pp 50-59. 2013.

[5] A. Abusukhon, Z. Mohammad, and M. Talib "A Novel Network Security Algorithm Based on Encrypting Text into a White-page Image". In proceedings of the World Congress on Engineering and Computer Science 2016 vol. 1, WCECS 2016, October 19-21, 2016, San Francisco, USA, 2016.

[6] A. Abusukhon, and B. Hawashin "A Secure Network Communication Protocol Based on Text to Barcode Encryption Algorithm" International Journal of Advanced Computer Science and Applications (IJACSA), vol. 6, pp. 64-70. 2015.

[7] S. Nithya, S., and E. George Dharma Prakash Raj "Survey on Asymmetric Key Cryptography Algorithms" Journal of Advanced Computing and Communication Technologies. vol.2, pp. 1-4, 2014.

[8] A. Abusukhon, M.N. Anwar, Z. Mohammad and B. Alghannam "A hybrid network security algorithm based on Diffie Hellman and Text-to-Image Encryption algorithm". Journal of Discrete Mathematical Sciences and Cryptography, 22(1), pp. 65-81.

[9] M. Saha, and D. RoyChowdhury "Provably Secure Key Establishment Protocol Using One-Way Functions" Journal of Discrete Mathematical Sciences & Cryptography. vol.12, pp. 139-158. 2009.

[10] R. Dutta, and R. Barua 2005 "Overview of Key Agreement Protocols". Cryptology ePrint Archive, pp.1-46. 2005.

[11] W. Diffie and M. Hellman "New directions in cryptography" Information Theory, IEEE Transactions on, vol. 22, pp. 644 – 654. 1976.

[12] N. Mohamed, A. Yasmine, E. Galal, and A. Amr "New Authenticated Key Agreement Protocols" in proceeding of International Multi Conference of Engineering and Computer Scientists, vol 1. Hong Kong, Pp. 58-63. 2013.

[13] H.M. Elkamchouchi, Y.A. Saleh, and A.M. Sary "New Authenticated Key Agreement Protocols". International Conference on Computer Engineering & Systems (ICCES), Cairo, 29 November 2011-1 December 2011, pp.58-63. 2011.

[14] M.S. Anoop "Elliptic curve cryptography – an implementation tutorial".Technical Report, Tata Elxsi Ltd, 2007.

[15] R. Ahirwal, M. Ahke "Elliptic Curve Diffie-Hellman Key Exchange Algorithm for Securing Hypertext Information on Wide Area Network" International Journal of Computer Science and Information Technologies (IJCSIT), vol. 4, pp. 363 – 368. 2013.

[16] L. LAW, A. MENEZES, M. QU, J. SOLINAS, and S. VANSTONE "An efficient protocol for authenticated key agreement". Tech. Rep. CORR 98-05, Department of C&O, University of Waterloo. [available online from http://grouper.ieee.org/groups/1363/] 1998.

[17] S. BURTON, JR. Kaliski "An Unknown Key-Share Attack on the MQV Key Agreement Protocol", ACM Transactions on Information and System Security, vol. 4, pp. 275–288, 2001.

[18] L. Law, A. Menezes, Qu. Minghua, J. Solinas, and S. Vanstone " An Efficient Protocol For Authenticated Key Agreement. Designs, Codes and Cryptography". vol 28, pp. 119–134. 2003.

[19] F. Hao "On Robust Key Agreement Based on Public Key Authentication" FC 2010, LNCS 6052, IFCA/Springer-Verlag Berlin Heidelbergpp. Pp. 383–390, 2010.

[20] A. Menezes, P. Van Oorschot, S. Vanstone "Handbook of Applied Cryptography". 1st ed., CRC Press, Boca Raton 1996.

[21] L.Hwang, C.C. Lee, and M.S. Hwang. "A n^2 + n MQV Key Agreement Protocol" International Arab Journal of Information Technology (IAJIT). vol 2, 2013.

[22] Z. Mohammad, Y.C. Chen, C.L. Hsu, and C.C., Lo "Cryptanalysis and enhancement of two-pass authenticated key agreement with key confirmation protocols". IETE Technical Review, vol. 27, pp. 252-265. 2010.

[23] Z. Mohammad, C.-L. Hsu, Y.-C. Chen, and C.-C. Lo, "Cryptanalysis of a secure and efficient three-pass authenticated key agreement protocol based on elliptic curves," Journal of Internet Technology, vol. 14, no. 2, pp. 247–250, 2013.

[24] B.Hawashin and A.Mansour "An Efficient Agent-Based System to Extract Interests of User Groups". In Proceedings of the World Congress on Engineering and Computer Science (Vol. 1).

[25] B.Hawashin, F. Farshad, and M.T.Traian. "A privacy preserving efficient protocol for semantic similarity join using long string attributes." In *Proceedings of the 4th International Workshop on Privacy and Anonymity in the Information Society*, p. 6. ACM, 2011.

[26] B.Hawashin, A. Mansour, T.Kanan, F.Fotouhi "An efficient cold start solution based on group interests for recommender systems". In Proceedings of the First International Conference on Data Science, E-learning and Information Systems (p. 26). ACM. 2018.

[27] A.M. Mansour, M.A. Obaidat and B. Hawashin "Elderly people health monitoring system using fuzzy rule based approach". International Journal of Advanced Computer Research, 4(4), 904, 2014.

[28] M. Lafi and A. Abdel Qader, "A novel dynamic integrated model for automated requirements engineering process," International Journal of Computer Applications in Technology, vol. 56, no. 4, pp. 292–300, 2017.

[29] M. Lafi and A. Abdel Qader, "A novel automated requirements prioritization and selection model based on requirements weights and stakeholder importance," International Journal on Information Technology (IREIT), vol. 4, no. 4, pp. 105–109, 2016.