# Improvement of a Security Enhanced One-time Mutual Authentication and Key Agreement Scheme

Tashev Komil Akhmatovich, Khudoykulov Zarif Turakulovich, Arzieva Jamila Tileubayevna

***Abstract***: *So far, many one-time password based authentication schemes have been proposed; however, none is secure enough. In 2004, W.C.Ku proposed hash-based strong-password based authentication scheme without using smart card that is vulnerable to the password guessing attack, not achieving mutual authentication and key agreement. In this paper, we propose a new improved version of Ku's scheme that is eliminated these weaknesses.*

***Keywords***: *authentication, one-time, smart card, key exchange, password, mutual.*

## I. INTRODUCTION

Password-based authentication is one of the most common, simple and easy-to-use authentication methods. Since static passwords are most often used in password-based authentication, it is vulnerable to the guessing attack, dictionary-based attack and replay attack. However, there are one-time password authentication methods that are used to eliminate these issues and are now widely used in remote user authentication. In particular, the hash-based authentication schemes are based on one-way functions and challenge-response mechanism and they are often preferred because of their ease of use and low computing power.

One of the first hash-based authentication methods is Lamport's hash-chain method that introduced in 1981 [1]. Although authentication methods based on Lamport schema are simple, it requires multiple hashing operations for users during each authentication session. In turn, it requires much time and device capacity for calculation [2]. The hash-based authentication schemes can be divide into two groups depending on whether or not they are using a token to storage data: scheme used smart card (CINON [3], OSPA-1 [4], ROSI [5], CLH [6], SPAPA [7], SAS-3 [8], JAN [9] and scheme using without smart card (Ku [10], SAS [11], OSPA [12]).

The SAS (Simple and Secure) authentication method developed by M.Sandirigama et al., does not require data storage, high computation  and transmission over network. In addition, it is not vulnerable to the "man in the middle" attack [11]. However, SAS protocol is vulnerable replay and DOS attacks [12].

Although these issues have been eliminated in OSPA (Optimal Strong-Password Authentication) [12] and in the Revised version of SAS protocol [15], implementation on small-scale devices is difficult (for example, in SAS-R authentication method, 5 times hashing operations are required for each session). Although SAS-2 protocol and its "challenge-response" version [13] have been developed to minimize the number of hashing iteration and to provide mutual authentication, they still require storing of some information, such as, random number. These problems can also be meted in a number of Lamport and SAS like authentication methods [24]-[25].

In this paper, we first propose the following seven security requirements for evaluating a password authentication scheme. Each requirement is an important and independent requirement for a new password authentication scheme.

1. *Resistance to Denial-of-Service Attack.* In this attack, attacker causes to fail legitimate user in his next attempt to authenticate by exchanging information belonging to him. After that, the legitimate user can no longer pass the authentication process.
2. *Resistance to Forgery Attack.* The attacker modifies the connection to falsify the legitimate user and thereby enter to the system.
3. *Resistance to Man-in-the-middle Attack.* In this case, the attacker will be able to log in a system through the connecting to communication line in two consecutive sessions. In this case, it is important to protect the communication line from listening, modification and damage.
4. *Resistance to Replay Attack.* In this attack, the attacker will deceived or discredit other legitimate users by reusing the protocol information.
5. *Resistance to Guessing Attack.* An entropy of password used for authentication must be high. In this attack, an attacker tries to crack the password in offline or online situation.
6. *Resistance to Stolen Verifier.* The attacker discredits the legitimate user by stealing password-verifiers (for example, hashed passwords) from the server.
7. *Providing Mutual Authentication.* The user and the server can be able to authenticate each other. That is, not only does the server authenticate the user, it also allows the user to authenticate the server.

The security requirements mentioned above are the most important ones for one-time password based authentication protocols. In addition, there are some types of attacks that are not applicable to all one-time password based authentication protocols.

**Revised Manuscript Received on October 30, 2019.**
* Correspondence Author

**Tashev Komil Akhmatovich***, Vice Rector for Scientific Affairs, Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Tashkent, Uzbekistan. Email: k.tashev@tuit.uz

**Khudoykulov Zarif Turakulovich**, head of the Cryptology department, Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Tashkent, Uzbekistan. Email: zarif.xudoyqulov@mail.ru

**Arzieva Jamila Tileubayevna***, Department of Applied Mathematics Karakalpak State University, Uzbekistan. Email: jamka-1980@mail.ru

*Retrieval Number: L37611081219/2019©BEIESP*
*DOI:10.35940/ijitee.L3761.1081219*
*Journal Website: www.ijitee.org*

*Published By:*
*Blue Eyes Intelligence Engineering &*
*Sciences Publication*

When evaluating the effectiveness of one-time password based authentication protocols, the following factors are important:

- number of hashing and stored data on the server side;
- number of hashing and stored data on the client side;
- an amount of data transferred from client to server for an authentication session.

The rest of the paper is organized as follows. In Section 2, we review W.C.Ku's scheme and some attacks against his scheme are described. After that, in Sections 3, we propose a new scheme and analyze its security. After that, the security and performance comparisons are presented in Section 4, and we conclude the paper in Section 5.

## II. REVIEW AND SECURITY WEAKNESSES OF THE KU'S PROTOCOL

Strong-password based authentication method developed by W.S.Ku does not require securely storing any information from client except password.

Following notations are used in Ku's protocol:

- $U$ denote the user and $S$ denote the server.
- $h()$ represents a cryptographic hash function. $h()$ denote the data $m$ is hashed one time and $h^2(m)$ is hashed twice, that is, $h^2(m) = h\big(h(m)\big)$.
- $P$ represents $U$'s strong password.
- $N$ denote a sequence number starting from 1 since $U$'s initial registration and incremented in authentication process.
- $K_S$ denote $S$'s secret key commonly used for generating a unique storage key for each user.
- $T$ denote the latest time $U$ initially registers or re-registers to $S$.
- Notation $\oplus$ denotes the bitwise $XOR$ operation and $||$ denotes the
- concatenation operation.
- Notation $A \rightarrow B{:}\,X$ means $A$ sends $X$ to $B$ through a common communication channel.
- Notation $A \Rightarrow B{:}\,X$ means $A$ sends $X$ to $B$ through a secure communication channel.

Ku's scheme involves two process, the registration process and the login process, which can be described as in the following.

***Registration process.*** This process is invoked whenever $U$ initially registers or re-registers to $S$.

***Step 1.*** $U \longrightarrow S$: registration request.

***Step 2.*** $S \Rightarrow U{:}\,N, T$.

$S$ sets $T$ to the value of his current timestamp. If it is $U$'s initial registration, $S$ sets $N$ to 1. Otherwise, $S$ sets $N = N + 1$. Next, $S$ sends $N$ and $T$ to $U$ through an secure channel.

***Step 3.*** $U \Rightarrow S{:}\,h^2(S \parallel P \parallel N \parallel T)$.

$U$ computes the verifier $h^2(S \parallel P \parallel N \parallel T)$ and then sends it to $S$ through an secure channel.

***Step 4.*** $S \longrightarrow U$: Information about successfully registration.

$S$ computes the storage key $K_U^{(T)} = h(U \parallel h(K_S \parallel T))$ to store $U$'s data and then computes the sealed verifier $sv^{(N)} = h^2(S \parallel P \parallel N \parallel T) \oplus K_U^{(T)}$. $S$ saves $sv^{(N)}$, $N$ and $T$ in password file.

***Login process.*** This process is invoked whenever $U$ logins $S$.

***Step 1.*** $U \longrightarrow S$: login request.

***Step 2.*** $S \longrightarrow U{:}\,r, n, t$.

$S$ selects randomly nonce $r$ and send it with $n = N$ and $t = T$ to $U$.

***Step 3.*** $U \longrightarrow S$: $c_1, c_2, c_3$.

$U$ computes following and send them to $S$:

$$c_1 = h^2(S \parallel P \parallel n \parallel t) \oplus h(S \parallel P \parallel n \parallel t)$$

$$c_2 = h^2(S \parallel P \parallel n + 1 \parallel t) \oplus h(S \parallel P \parallel n \parallel t)$$

$$c_3 = h(h^2(S \parallel P \parallel n + 1 \parallel t) \parallel r)$$

***Step 4.*** $S \longrightarrow U$: Information about whether or not an user is authenticated.

$S$ computes $K_U^{(T)} = h(U \parallel h(K_S \parallel T))$ and extract $h^2(S \parallel P \parallel n \parallel t)$ from $sv^{(N)}$, that is:

$$h^2(S \parallel P \parallel n \parallel t) = sv^{(N)} \oplus K_U^{(T)}.$$

Then, $S$ computes $u_1$ and $u_2$ as follow:

$$u_1 = c_1 \oplus h^2(S \parallel P \parallel n \parallel t) = h(S \parallel P \parallel n \parallel t)$$

$$u_2 = c_2 \oplus u_1 = h^2(S \parallel P \parallel n + 1 \parallel t).$$

If $h(u_1) = h^2(S \parallel P \parallel n \parallel t)$ and $h(u_2 \parallel r{=}c3$, then $S$ authenticates $U$. Otherwise, $S$ rejects $U$'s login request and terminates this session.

After user $U$ is authenticated, $S$ computes following for next session:

$$sv^{(n+1)} = u_2 \oplus K_U^{(t)} = h^2(S \parallel P \parallel n + 1 \parallel t) \oplus K_U^{(t)}$$

And $S$ replaces $sv^{(N)}$ with $sv^{(n+1)}$and sets $N = n + 1$ for $U$'s next login. The value of $T$ is unchanged: $T = t$.

***Security analysis of Ku's protocol.*** This protocol is analyzed by many researchers [22, 23] and they implement attack on the protocol in case of verifier, $h^2(S \parallel P \parallel n \parallel t)$ is stolen. However, to get verifier, attacker needs to know the secret key of $S$ (that is, $K_S$). If the attacker knows the secret key of $S$, then it will cause more serious problems than breaking the protocol. Nevertheless, without getting the verifier, Ku's protocol has following security weakness:

1. Ku's protocol is weakness to guessing attack when poor passwords are used. That is, attacker can try to check his guessing password $P$ validity by handled $c_1$ hash value as follow:

$$c_1 = h^2(S \parallel P \parallel n \parallel t) \oplus h(S \parallel P \parallel n \parallel t)$$

If attacker found valid password $P$ that satisfy above equation, then entity protocol is broken.

2. In addition, Ku's protocol cannot provide mutual authentication. That is, user cannot check server validity.

3. Ku's protocol perform only one-way authentication without any key agreement.

## III. IMPROVED VERSION OF KU'S PROTOCOL

Improved version of Ku's protocol consist of three process: *registration process, login and key agreement process* and *password changing process.*

***Registration process.*** This process is invoked whenever $U$ initially registers or re-registers to $S$.

***Step 1.*** $U \longrightarrow S$: Registration request.

***Step 2.*** $S \Rightarrow U{:}\,N, T$.

$S$ sets $T$ to the value of his current timestamp. If it is $U$'s initial registration, $S$ sets $N$ to 1. Otherwise, $S$ sets $N = N + 1$. Next, $S$ sends $N$ and $T$ to $U$ through an secure channel.

***Step 3.*** $U \Rightarrow S$: $h^2(S \parallel P + K_N \parallel N \parallel T)$.

$U$ computes the verifier $h^2(S \parallel P + K_N \parallel N \parallel T)$ and then sends it to $S$ through an secure channel. There is, $K_N$ – nonce that generated by user $U$ and securely saved.

***Step 4.*** $S \longrightarrow U$: Information about successfully registration

$S$ computes the storage key $K_U^{(T)} = h(U \parallel h(K_S \parallel T))$ to store $U$'s data and then computes the sealed verifier $sv^{(N)} = h^2(S \parallel P + K_N \parallel N \parallel T) \oplus K_U^{(T)}$. $S$ saves $sv^{(N)}$, $N$ and $T$ in password file.

***Login and key agreement process.*** This process is invoked whenever $U$ logins $S$.

***Step 1.*** $U \longrightarrow S$: Login request.

***Step 2.*** $S \longrightarrow U$: $r, n, t$.

$S$ selects randomly nonce $r$ and send it with $n = N$ and $t = T$ to $U$.

***Step 3.*** $U \longrightarrow S$: $c_1, c_2, c_3$.

$U$ generates $K_{N+1}$ and computes following and send them to $S$:

$$c_1 = h^2(S \parallel P + K_N \parallel n \parallel t) \oplus h(S \parallel P + K_N \parallel n \parallel t)$$

$$c_2 = h^2(S \parallel P + K_{N+1} \parallel n + 1 \parallel t) \oplus h(S \parallel P + K_N \parallel n \parallel t)$$

$$c_3 = h(h^2(S \parallel P + K_{N+1} \parallel n + 1 \parallel t) \parallel r)$$

***Step 4.*** $S \longrightarrow U$: $c_4$ and information about whether or not an user is authenticated.

$S$ computes $K_U^{(T)} = h(U \parallel h(K_S \parallel T))$ and extract $h^2(S \parallel P + K_N \parallel n \parallel t)$ from $sv^{(N)}$, that is:

$$h^2(S \parallel P + K_N \parallel n \parallel t) = sv^{(N)} \oplus K_U^{(T)}.$$

Then, $S$ computes $u_1$ and $u_2$ as follow:

$$u_1 = c_1 \oplus h^2(S \parallel P + K_N \parallel n \parallel t) = h(S \parallel P + K_N \parallel n \parallel t)$$

$$u_2 = c_2 \oplus u_1 = h^2(S \parallel P + K_{N+1} \parallel n + 1 \parallel t).$$

If $h(u_1) = h^2(S \parallel P + K_N \parallel n \parallel t)$ and $h(u_2 \parallel r) = c_3$, then $S$ authenticates $U$. Otherwise, $S$ rejects $U$'s login request and terminates this session.

After user $U$ is authenticated, $S$ computes followings for next session:

$$sv^{(n+1)} = u_2 \oplus K_U^{(t)} = h^2(S \parallel P + K_{N+1} \parallel n + 1 \parallel t) \oplus K_U^{(t)}$$

And $S$ replaces $sv^{(N)}$ with $sv^{(n+1)}$and sets $N = n + 1$ for $U$'s next login. The value of $T$ is unchanged: $T = t$.

After successfully authentication, server $S$ computes session key $K_C$ as follow:

$$K_C = h(U \parallel u_1).$$

$S$ calculates following to authenticated by user $U$ and send it to $U$:

$$c_4 = h(U \parallel u_2).$$

***Step 5.*** $U \longrightarrow S$: Information about whether or not an server is authenticated.

If equation $h(U \parallel h^2(S \parallel P + K_{N+1} \parallel n + 1 \parallel t) =?\, c_4$ is true, then $U$ authenticates $S$ and calculates session key $K_C$ as follow:

$$K_C = h(U \parallel h(S \parallel P + K_N \parallel n \parallel t).$$

Finally, user $U$ saves $K_{N+1}$ instead of $K_N$ for next authentication request.

***Password changing process.*** Password changing process is one the main requirements from one-time password based authentication methods and in generally, this process is not done easily as in static password based authentication. Therefore, in many one-time password based authentication methods, such as Ku's protocol, formalization of password changing process is not given. Proposed improved version of Ku's protocol has password changing process and it is done as follow:

***Step 1.*** $U \longrightarrow S$: Password changing request.

***Step 2.*** $S \longrightarrow U$: $r, n, t$.

$S$ generates nonce $r$ and send it to $U$ with $n = N$ and $t = T$ from password file.

***Step 3.*** $U \longrightarrow S$: $c_1, c_2, c_3$.

$U$ generates random number $K_{N+1}$ and new password $P'$ and calculates following:

$$c_1 = h^2(S \parallel P + K_N \parallel n \parallel t) \oplus h(S \parallel P + K_N \parallel n \parallel t)$$

$$c_2 = h^2(S \parallel \acute{P} + K_{N+1} \parallel n + 1 \parallel t) \oplus h(S \parallel P + K_N \parallel n \parallel t)$$

$$c_3 = h(h^2(S \parallel \acute{P} + K_{N+1} \parallel n + 1 \parallel t) \parallel r)$$

$U$ send $c_1, c_2, c_3$ to the server $S$.

***Step 4.*** $S \longrightarrow U$: information about password changed and $c_4$.

$S$ calculates $K_U^{(T)} = h(U \parallel h(K_S \parallel T))$ and extract $h^2(S \parallel P + K_N \parallel n \parallel t)$ from $sv^{(N)}$as follow :

$$h^2(S \parallel P + K_N \parallel n \parallel t) = sv^{(N)} \oplus K_U^{(T)}.$$

Then, $S$ calculates $u_1$ and $u_2$ as follow:

$$u_1 = c_1 \oplus h^2(S \parallel P + K_N \parallel n \parallel t) = h(S \parallel P + K_N \parallel n \parallel t)$$

$$u_2 = c_2 \oplus u_1 = h^2(S \parallel P' + K_{N+1} \parallel n + 1 \parallel t).$$

If $h(u_1) = h^2(S \parallel P + K_N \parallel n \parallel t)$ and $h(u_2 \parallel r) = c_3$ are valid, then $S$ authenticates $U$ and calculates password verifier based on new password as follow:

$$sv^{(n+1)} = u_2 \oplus K_U^{(t)} = h^2(S \parallel P' + K_{N+1} \parallel n + 1 \parallel t) \oplus K_U^{(t)}$$

And, for login process, $S$ changes $sv^{(N)}$ with $sv^{(n+1)}$ and sets $N = 1$. Current time stamp is not changed.

$S$ calculates $c_4$ as follow to authenticated by $U$ and send it with information about successfully password changed:

$$c_4 = h(U \parallel u_2).$$

If $h(U \parallel h^2(S \parallel P + K_{N+1} \parallel n + 1 \parallel t) =?\, c_4$ is valid, then $U$ authenticates $S$ and informed about password changed.

## IV. SECURITY AND PERFORMANCE CONSIDERATION

In this section, we will briefly demonstrate the improved scheme is secure against the replay attack, offline-guessing attack, stolen verifier attack, the denial of service attack, forgery attack, and man-in-the-middle attack. Furthermore, we will also show that the improved scheme is effective to perform.

*Resistance to Denial-of-Service Attack.* To prevent the denial-of-service attack, $S$ has to make sure that the computed $u_2$, which will be used $U's$ next verifier, is authentic. Since $c_3$can protect the integrity of $c_1$ and $c_2$, which are used to compute $u_2$, any unauthorized modification on $c_1$, $c_2$ and $c_3$ will be detected by $S$. As the attacker can not disable $U's$ account, the improved scheme can resist the denial of service attack.

*Resistance to Forgery Attack.* To mount a forgery attack on the improved scheme, the attacker must generate the authentication message corresponding to the given $n$ and $r$. Since the attacker knows neither $P$ and $K_N$ or $h(S \parallel P + K_N \parallel n \parallel t)$, he can not produce the correct $\{c_1, c_2, c_3\}$ that will be accepted by $S$. Hence, the proposed scheme can resist the forgery attack.

*Resistance to Man-in-the-middle Attack.* The man-in-the-middle attack takes places when the attacker, who sits between the client and the server, tries to cheat either side by eavesdropping on the transmission data, modifying the message and then relaying the modified message. The improved scheme is resistance to man-in-the-middle attack.

Modifying $c_1$ of Step 3 will cause the server to fail on recovering the correct $h(S \parallel P + K_N \parallel n \parallel t)$ from $c_2$, and modifying $c_2$ of Step 3 also causes the server to fail on recovering the correct $h^2(S \parallel P' + K_{N+1} \parallel n+1 \parallel t)$. The server will reject the request when it cannot recover the correct $h^2(S \parallel P' + K_{N+1} \parallel n+1 \parallel t)$ to check $h\big(h^2(S \parallel$ *Ṕ+KN+1‖n+1‖t)‖r* equals *c3*. The attacker must know both $h(S \parallel P + K_N \parallel n \parallel t)$ and $h^2(S \parallel P' + K_{N+1} \parallel n+1 \parallel t)$ before he can modify both $c_1$ and $c_2$ simultaneously while maintaining the validity of the modified data, which is infeasible because of only the right user who inputs the correct password into the tamper – resistant device can derive both of the secret values.

*Resistance to Replay Attack.* Suppose that $N = n$ and the attacker has captured all $U's$ past authentication messages $\{c_1^{(i)}, c_2^{(I)}, c_3^{(i)}\}$ for $i = 1, 2, \ldots, n-1$. Since $U's$ current verifier stored in $S$ is $h^2(S \parallel P + K_N \parallel n \parallel t)$, attacker can not login $S$ by using $\{c_1^{(i)} = h^2(S \parallel P + K_N \parallel i \parallel t) \oplus h(S \parallel P + K_i \parallel i \parallel t), c_2^{(i)}, c_3^{(i)}\}$, where $1 \le i \le n-1$. Alternatively, if the adversary replaces the transmitting $c_2^{(n)}\ and\ c_3^{(n)}$ with $c_2^{(i)}\ and\ c_3^{(i)}$, where $i = 1, 2, \ldots, n-1$, during $U's$ login, $S$ will detect this fraudulence because $h((c_2^{(i)} \oplus \left(c_1^{(n)} \oplus h^2(S \parallel P \oplus K_n \parallel n \parallel t)\right)) \oplus r^{(n)})$ does not equal $c_3^{(i)} = h(h^2(S \parallel P \oplus K_{n+1} \parallel n+1 \parallel t) \parallel r^{(i)}))$. Note that even if the attacker could fool $S$ into replacing $U's$ verifier $h^2(S \parallel P + K_n \parallel n \parallel t)$ with $h^2(S \parallel P + K_i \parallel i \parallel t)$, where $1 \le i \le n-1$, by some means, the attacker can not impersonate $U$ because $r^{(n)} \ne r^{(i)}$, which implies $h((c_2^{(i)} \oplus \left(c_1^{(n)} \oplus h^2(S \parallel P + K_n \parallel n \parallel t)\right)) \oplus r^{(n)}) \ne c_3^{(i)}$. Therefore, the improved scheme can resist the replay attack.

*Resistance to Guessing Attack.* Suppose that the attacker has captured all $U's$ past authentication messages: $c_1, c_2, c_3$. Even the attacker tries to knows $U's$ password $P$, he cannot use $c_1 = h^2(S \parallel P + K_N \parallel n \parallel t) \oplus h(S \parallel P + K_N \parallel n \parallel t)$ equation to verify his guessed password because of knowing nonce $K_N$ is required. Therefore, the improved scheme is resistance to offline guessing attack.

*Resistance to Stolen Verifier.* Suppose that $S's$ password file was compromised to the attacker, i.e, the attacker has obtained $'s$ $T(=t)$, $N(=n)$, and the sealed verifier $sv^{(N)} = h^2(S \parallel P + K_N \parallel N \parallel T) \oplus K_U^{(T)}$. Clearly, $t$ and $n$ are not secrets. As $sv^{(N)} = h^2(S \parallel P + K_N \parallel N \parallel T) \oplus K_U^{(T)}$, the attacker can derive $h^2(S \parallel P + K_N \parallel N \parallel T)$ from $sv^{(N)}$ only if he knows $K_U^{(T)} = h(U \parallel h(K_S \parallel T))$, which implies that he knows $K_S$. As assumed, $K_S$ is under strict protection, and therefore the improved scheme can resist the stolen verifier attack. In other words, if $K_S$ is top secret of the system and it is compromised, then the entity system could be under serious attack.

Furthermore, improved version of Ku's protocol compares to currents one based on above requirements and results are presented in Table 1.

**Table I: Comparison between the existing protocols and the improved protocol with respect to attacks**

| | R1 | R2 | R3 | R4 | R5 | R6 | R7 |
|---|---|---|---|---|---|---|---|
| Lamport scheme [1] | + | + | + | -[16] | + | + | - |
| CINON [3] | - | + | -[12] | - | + | - | - |
| PERM [14] | + | -[17] | -[8] | + | + | - | - |
| SAS [11] | -[12] | + | + | -[12] | -[5] | -[18] | - |
| SAS-1 [15] | -[5] | + | + | - | + | -[5] | - |
| OSPA [12] | + | -[15] | -[15] | + | - | -[18] | - |
| OSPA-1 [4] | -[19] | + | + | -[19] | + | + | - |
| ROSI [5] | -[20] | + | + | + | + | + | + |
| SAS-2 [13] | -[5] | -[8] | + | + | -[5] | -[5] | + |
| CLH [6] | -[17] | + | + | + | + | + | + |
| SPAPA [7] | + | + | -[21] | -[21] | + | -[21] | + |
| SAS-3 [8] | + | + | N/A | + | + | N/A | + |
| JAN [9] | N/A | + | + | + | + | + | + |
| Ku [10] | + | + | + | -[22] | -[23] | + | - |
| Improved Ku's scheme | + | + | + | + | + | + | + |

Table 2 summarize the performance of the existing protocols and improved version of Ku's protocol in the $i$ th authentication session. $h$ represents the hash value, $M$ is the maximum number of hash iterations, $V_i$ is the verifier in $i$ th authentication session and $L(x)$ represents the data length of $x$.

**Table- II: Evaluating the effectiveness of existing protocols and the improved protocol**

| | Server | | User | | User ✇Server | |
|---|---|---|---|---|---|---|
| | ***Hash iterations (times)*** | ***Data storages*** | ***Hash iterations (times)*** | ***Data storages*** | ***Transmission iterations*** | ***Transmission bulk*** |
| Lamport scheme [1] | 1 | $n$ | $M-n$ | $n$ | 1 | $L(H)$ |
| CINON [3] | 2 | $N_i, N_{i+1}$ | 5 | $E_i^1, E_{i+1}^1, M_n$ | 1 | $3L(H)$ |
| SAS [11] | 1 | $A, h^2(P \vert \vert N_n), N_n$ | 5 | $-$ | 2 | $L(A) + L(Ser.req)$ $+ 2L(h) + L(N_n)$ |
| SAS-1 [15] | 2 | $ID, V_i$ | 5 | $N_i$ | 1 | $L(ID) + 2L(H)$ |
| OSPA [12] | 4 | $A, h^2(P \oplus n), n$ | 9 | $-$ | 2 | $L(A) + L(Ser.Req)$ $+ 3L(H)$ |
| OSPA-1 [4] | 4 | $ID, h^2(P \oplus N)$ | 5 | Token $(K, N)$ | 1 | $L(ID) + 2L(H)$ |
| ROSI [5] | 3 $(mutual\ 6)$ | $ID, h^2(S \vert \vert N_i)$ | 4 | Token $(R, h(S \vert \vert N_i))$ | 1 | $L(ID) + 2L(H)$ |

| | | | | | | |
|---|---|---|---|---|---|---|
| SAS-2 [13] | $1\ (mutual\ 2)$ | $ID, V_i$ | $3\ (mutual\ 4)$ | $N_i$ | 1 | $L(ID) + 2L(H)$ |
| CLH [6] | $4\ (mutual\ 6)$ | $ID, h^2(PW_i \oplus N)$ | 4 | Token $(N, h(x\|\|ID))$ | $1\ (mutual\ 2)$ | $L(ID) + 4L(H)$ |
| SPAPA [7] | 5 | $A, h(A\|\|N\|\|P)$, $h^2(P \oplus N)$ | 7 | Token $(K, N)$ | 1 | $5L(H)$ |
| SAS-3 (2) [8] | $\frac{1+n}{2}n$ | $B, UID, P, n$ | $\frac{1+n}{2}n + n - c$ | $n$ | 2 | $2L(IUD) + 2L(k) + L(c) + L(Z)$ |
| JAN [9] | $m_{i+1} + 2$ | $ID, N_i, N_{i+1}, m_i, m_{i+1}, T, SV_U^{N_i}, K_{ID}^T$ | $m_i + 2m_{i+1} + 4$ | Token $(ID, N_i, N_{i+1}, m_i, m_{i+1}, T, SV_U^{N_i})$ | 2 | $L(ID) + L(log.req) + 3L(H)$ |
| Ku [10] | 4 | $sv^{(N)}, T, N$ | 6 | $-$ | 1 | $L(log.req) + 3L(H)$ |
| Improved Ku's scheme | 4 (6 mutual and key agreement) | $sv^{(N)}, T, N$ | 6 (7 mutual) | $K_N$ | 1 | $L(log.req) + 4L(H)$ |

## V. CONCLUSION

In this paper, we focused on hash overhead and improved Ku's scheme. The improved version of Ku's scheme performs a little hashing operations that the original one because of it has mutual authentication phrase with key agreement procedure. However, improved version Ku's scheme is resistant to replay and offline guessing attacks that breaks original Ku's scheme. On the other hand, these advantages was not seriously effected to effectiveness of improved version Ku's scheme.

## REFERENCES

1. L. Lamport, "Password Authentication with Insecure Communication", In: Comm. ACM, vol. 24, No 11, 1981, pp. 770-772.
2. Takasuke TSUJI. *A One-Time Password Authentication Method*. Master's thesis. January 31, 2003.
3. Shimizu A. A dynamic password authentication method using a one-way function //Systems and computers in Japan. – 1991. – T. 22. – №. 7. – pp. 32-40.
4. Lin C. W., Shen J. J., Hwang M. S. Security enhancement for optimal strong-password authentication protocol //ACM SIGOPS Operating Systems Review. – 2003. – T. 37. – №. 2. – pp. 7-12.
5. Chien H. Y., Jan J. K. Robust and simple authentication protocol//The computer journal. – 2003. – T. 46. – №. 2. – pp. 193-201.
6. Chen T. H., Lee W. B., Horng G. Secure SAS-like password authentication schemes //Computer Standards & Interfaces. – 2004. – T. 27. – №. 1. – pp. 25-31.
7. Mangipudi K. V., Katti R. S. A Hash-based Strong Password Authentication Protocol with User Anonymity //IJ Network Security. – 2006. – T. 2. – №. 3. – pp. 205-209.
8. Weragama N. S., Sandirigama M. SAS-3: A polynomial based strong password authentication protocol //2007 International Conference on Industrial and Information Systems. – IEEE, 2007. – pp. 41-46.
9. Jan M. S., Afzal M. Hash chain based strong password authentication scheme //2016 13th International Bhurban Conference on Applied Sciences and Technology (IBCAST). – IEEE, 2016. – pp. 355-360.
10. Ku W. C. A hash-based strong-password authentication scheme without using smart cards //ACM SIGOPS Operating Systems Review. – 2004. – T. 38. – №. 1. – pp. 29-34.
11. Sandirigama M., Shimizu A., Noda M. T. Simple and secure password authentication protocol (SAS) //IEICE Transactions on Communications. – 2000. – T. 83. – №. 6. – pp. 1363-1365.
12. Lin C. L., Sun H. M., Hwang T. Attacks and solutions on strong-password authentication //IEICE transactions on communications. – 2001. – T. 84. – №. 9. – pp. 2622-2627.
13. Tsuji T., Kamioka T., Shimizu A. Simple and secure password authentication protocol, ver. 2 (SAS-2) //ITE Technical Report 26.61. – The Institute of Image Information and Television Engineers, 2002. – pp. 7-11.
14. Shimizu A., Horioka T., Inagaki H. A password authentication method for contents communications on the Internet //IEICE transactions on communications. – 1998. – T. 81. – №. 8. – pp. 1666-1673.
15. Tsuji T., Shimizu A. An impersonation attack on one-time password authentication protocol OSPA //IEICE Transactions on Communications. – 2003. – T. 86. – №. 7. – pp. 2182-2185.
16. Chen L., Mitchell C. J. Comments on the S/KEY user authentication scheme //ACM Operating Systems Review. – 1996. – T. 30. – №. 4. – pp. 12-16.
17. Mangipudi K. V., Katti R. S. A Hash-based Strong Password Authentication Protocol with User Anonymity //IJ Network Security. – 2006. – T. 2. – №. 3. – pp. 205-209.
18. Chen C. M., Ku W. C. Stolen-verifier attack on two new strong-password authentication protocols //IEICE Transactions on communications. – 2002. – T. 85. – №. 11. – pp. 2519-2521.
19. Ku W. C., Tsai H. C., Chen S. M. Two simple attacks on Lin-Shen-Hwang's strong-password authentication protocol //ACM SIGOPS Operating Systems Review. – 2003. – T. 37. – №. 4. – pp. 26-31.
20. Ku W. C., Tsai H. C., Tsaur M. J. A Common Weakness of Password Authentication Schemes Requiring Synchronous Update of Stored Data //Proceedings of the 2004 International Computer Symposium, Taiwan. – 2004. – pp. 849-852.
21. Mitchell C. J., Ng S. L. Comments on the security of the SPAPA strong password authentication protocol. – 2007.
22. Kim M., Koç C. K. A Simple Attack on a Recently Introduced Hash-based Strong-password Authentication Scheme //IJ Network Security. – 2005. – T. 1. – №. 2. – pp. 77-80.
23. Kumar M. On the security vulnerabilities of a hash based strong password authentication scheme //Organization. – 2009. – T. 4. – pp. 9.
24. Jo H. S., Youn H. Y. A secure user authentication protocol based on one-time-password for home network //International Conference on Computational Science and Its Applications. – Springer, Berlin, Heidelberg, 2005. – pp. 519-528.
25. N. Haller Bellcore, "The S/KEY One-Time Password System", Network Working Group, February 1995.

## AUTHORS PROFILE

**Tashev Komil Akhmatovich** received his MS degree in the Department of Computer systems from Tashkent state technical university, Uzbekistan, in 2005. He received PhD degree in 2010. His research interests include data protecting techniques, monitoring, image recognition, network security.

**Khudoykulov Zarif Turakulovich** received his MS degree in the Department of information security from Tashkent university of information technologies, Uzbekistan, in 2013. He received PhD degree in 2018. His research interests include data protecting techniques, cryptology, image recognition.

**Arzieva Jamila Tileubayevna** received his MS degree in the Department of Applied Mathematics Karakalpak State University, Uzbekistan, in 2007. She is PhD student. Her research interests include data protecting techniques, cryptography, and access control.