ОГЛАВЛЕНИЕ

[1 An anonymous authenticated key-agreement scheme for multi-server infrastructure 2](#_Toc162519872)

[1.1 Our contribution 2](#_Toc162519873)

[1.2 Preliminaries 2](#_Toc162519874)

[1.2.1 Hash functions 2](#_Toc162519875)

[1.3 Formal security analysis 3](#_Toc162519876)

[1.4 Performance analysis 4](#_Toc162519877)

[1.5 Chebyshev chaotic maps 4](#_Toc162519878)

# An anonymous authenticated key-agreement scheme for multi-server infrastructure

## Our contribution

An anonymous three factor authentication protocol is introduced in this paper and the authentication of users with the help of biometric impression is enhanced. We encompass our contributions as follows.

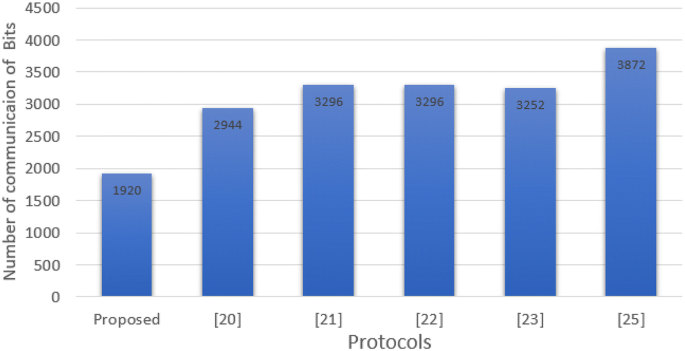
1. First, we introduce an ECC based three-factor user authenticated key-agreement protocol;
2. Second, if smart card can be forged by an adversary, then the environment of user cannot be secure. In our introduced protocol, the verification of biometric impression of users can be done by the client as well as by the server; in some specific applications it can provide security protection for specific requirements. RC and server have separate responsibilities, as RC is involved in authentication phase. RC retains the privacy of registration and server validates the client for further service providing; it can make the protocol more scalable for multi-server the architecture;
3. Finally, our protocol offers the mutual authentication for each pair of three participants (server, user and RC) for providing strong protection by identifying as possible replay messages.

Figure 1 - Analysis of communication cost between proposed and related protocols

## Our contribution

The hash functions, elliptic curve cryptography, adversarial model which is used in this paper are stated in this section. Whereas, Table 1 is presenting the common notations, used in rest of the article.

### Hash functions

By taking an input string O = H (String) of random size, a fixed size output is generated by hash. Generated output is called hash code. A little change in the value of string can cause a huge difference. Whereas, a secure one-way hash function has following specifications:

* If the string is described, it is easy to find O = H (String);
* It is impossible to find out the string, if O = H (String) is illustrated;
* It is mundane task to distinguish input of String1 and String2 so that H(String1) = H(String2).

## Formal security analysis

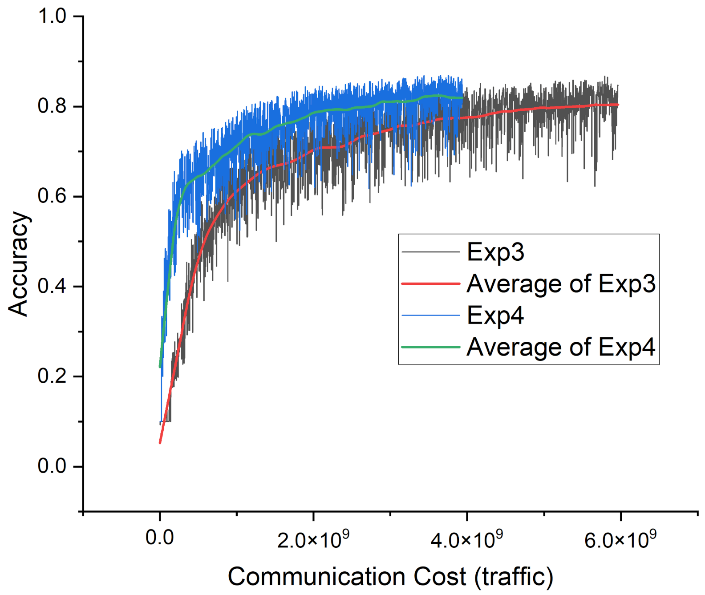
We have described model of security for presented protocol in this section. Furthermore, using given model of security the presented protocol is proved safe against known attacks. At the end, the proposed protocol is described to fulfill all the necessary requirements that relates to the security of the presented protocol.

Figure 2 - Cost of communication

Theorem THM1 Consider Di as a uniformly distributed dictionary consists of various possible passwords. |D| denotes the size of Di. Consider A as an adversary against semantic security within a time bound t. Suppose a ECCDH problem stands, then we have Di is considered as evenly distributed dictionary which consists of numerous passwords that can be possible. The size of Di is denoted by |D|. A is considered as an adversary against syntactic security in a time bound t. If a ECCDH problem occurs, then we have

|  |  |
| --- | --- |
|  | (1) |

where the possibility of solving the ECCDH problem by A, is denoted by AECCDH ­. The number of Execute, Random-oracle and Send query are {qexe, qhsh, qsnd}, respectively.

Proof In order to give the proof of Theorem THM1, six composite games are considered from game G1 to G6. The game will be started where the real attack is simulated and a game will be ended where adversary A has no advantage. The possibility of successfully guessing the random bit b in test-query by A is denoted by Suci for each game Gi, where 1≤i≤6.

|  |  |
| --- | --- |
| , | (2) |

Where – adversare;

Suc1 – possibility of successfully guessing;

Pr(Suc1) – per the definition of Suc1.

GAME G1: In this random oracle model, the real attacks are simulated with the help of this game. In game G1, every instance like Uu, Sj and RC will be modeled as authentic executions. As per the definition of Suc1, we get following equation.

## Performance analysis

Table 1 - Comparison of security parameters

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Scheme: | Proposed | Liao and Wang | Hsiang and Shih | Lee et al. | Chen and Lee | Irshad et al. |
| Immune to smart card stolen attack | Yes | Yes | Yes | Yes | No | Yes |
| Immune to trace attack | Yes | Yes | Yes | Yes | No | Yes |
| Immune to impersonation attack | Yes | No | No | No | No | Yes |
| Immune to KCI attack | Yes | Yes | No | Yes | No | Yes |

## Chebyshev chaotic maps

The chaotic map-based authentication protocols can be seen in the research literature and these Chaotic-encryption-based techniques are still being adopted as a tradeoff between security and computational cost. We can see few chaotic map variants, i.e., symmetric, asymmetric, and one-way hash functions, as being used in cryptography; however, most of the chaotic map-based techniques are following symmetric cryptosystems. For better understanding, some of the properties of Chebyshev polynomial and chaotic maps are defined as under:

Definition 1 To describe the first property of Chebyshev polynomial, we assume n as an integer, and a variable x of the interval [−1, 1]. While, we define the Chebyshev polynomial Tn(x): [−1, 1] → [−1, 1] as

,

The first few Chebyshev polynomials are listed as below (3) :

|  |  |
| --- | --- |
|  | (3) |