# DS-NIZKP: A ZKP-based Strong Authentication using Digital Signature for Distributed Systems

**Article** *in* International Journal of Computer Science and Information Security, · June 2018

CITATIONS
2

READS
574

**4 authors:**

Aline Z.Tsague

**4** PUBLICATIONS **5** CITATIONS

SEE PROFILE

Elie Fute Tagne
Université de Dschang

**46** PUBLICATIONS **85** CITATIONS

SEE PROFILE

Adnen El Amraoui
École Centrale de Lille

**71** PUBLICATIONS **282** CITATIONS

SEE PROFILE

Emmanuel Tonye
University of Yaounde I

**247** PUBLICATIONS **825** CITATIONS

SEE PROFILE

# DS-NIZKP: A ZKP-based Strong Authentication using Digital Signature for Distributed Systems

Aline Z. Tsague [1,2], Elie T. Fute [1,2], Adnen EL AMRAOUI [3], Emmanuel Tonye [4]

[1] *University of Buea, Cameroon*
linetsague@yahoo.fr

[2] *University of Dschang, Cameroon*
eliefute@yahoo.fr

[3] *University of Orléans, France*
elamraoui.adnen@gmail.com

[4] *University of Yaounde 1, Cameroon*
tonyee@hotmail.com

***Abstract*— In most networks and distributed systems, security has always been of a major concern and authentication is the core issue as it provides protection from unauthorized use and ensures proper functioning of the system. This paper investigates and proposes DS-NIZKP, an approach for authenticating users by three factors, (namely password, smart-card and biometrics) based on the concept of Zero Knowledge Proof (ZKP), so that no sensitive information can be revealed during a communication. The proposal employs the concept of digital signature (DS) to authenticate the identity of the sender or the signer within a single communication. Given that DS employs asymmetric encryption, a one-way hash of the user's identity is created then signed using the private key. Hashing prevents from revealing information about the user while signing provides authentication, non-repudiation and integrity. This approach not only saves time since just a single message between the prover and the verifier is necessary but also defends privacy of the user in distributed systems.**

***Keywords- Authentication; Digital Signature; Distributed System; Security; Three-factor authentication; Zero Knowledge Proof (ZKP)***

## I. INTRODUCTION

A distributed system is a collection of autonomous computational entities interconnected by a communication network. Each entity (also called node) in a distributed system has its own computing resources and is able to communicate with others in the system. A fundamental concern in building a secure distributed system is authentication of local and remote entities in the system [1]. Basically, authentication refers to a process in which a node (or user) provides some form of proof on his identity in order to access the system, under which he had previously registered for an account. This consists in proving that he is the same user who registered the account. In simple terms, authentication is identification plus verification [1]. A proof of identity can be any piece of information that an authentication server accepts: something users have in their possession, something they know or something they are [2]. The security of local or remote authentication mechanisms mostly relies on one of or the combination of three previous factors: something users know (like password), something users have (like smart card), and something users are (biometric characteristics like fingerprint). Also it is important that no authentication information leaks while authenticating a user in order to avoid a third party to impersonate a legitimate user. Zero knowledge protocols can be used to improve on this. In ZKP, a prover will try to demonstrate knowledge of a certain secret to a verifier without revealing any information whatsoever about the proof itself, except of course for the fact that it is indeed a valid one [3]. This paper focuses on the authentication issue and proposes a ZKP based authentication mechanism in a distributed system environment.

The rest of this paper is organized as follows: section 2 reviews related works on various authentication aspects in distributed systems. Section 3 presents a brief review of the cryptographic concepts that are relevant to the construction of our authentication scheme. In Section 4, we propose an authentication framework for distributed systems (the DS-NIZKP-based authentication). We first discuss the system setup and give an overview of the protocol before elaborating the details in each phase of the authentication. Section 5 assesses and provides some security proofs in relation with the proposal. Finally, conclusion remarks and perspectives are given in section 6.

## II. RELATED SURVEY

Several works have addressed the issue of user authentication in distributed environments since it is an essential factor to the security of data and communications among users. As a matter of fact, due to a wide spread use of mobile devices, a user can access various location based services or communicate with peer users almost everywhere he or she goes, through dynamically and temporarily formed networks [4]. Typically, user authentication is based on one or more of something users know, something users have and something they "are" [2] [5] [6]. Something user knows refers to password or PIN, social security number, mother's maiden name, pet's name, a picture, etc. Something uses have include physical key, proximity card, RSA SecureID token, smart card



https://sites.google.com/site/ijcsis/
ISSN 1947-5500

or credit card, SecureNet token, Cell phone, etc. Finally something they are typically refers to biometrics (face, fingerprint, voiceprint, iris, etc).

The most early authentication mechanisms (also known as password-based authentication) were solely based on password [5] [6]. Even though passwords are still efficient and relatively easy to implement and manage, they have shown limits in some cases (guess of passwords in relatively short time by simple dictionary attack when simple human generated passwords are used). Due to these concerns, hardware authentication tokens (such as smart-card-based password authentication) were introduced in order to strengthen the security in user authentication [5]. Another authentication mechanism is biometric authentication [2] [6] [7], which mainly employs measurable human characteristics such as fingerprint, voiceprint and iris scan to identify users. Thus multiple-factor authentication has become one of the most common authentication mechanisms given that it combines two or more factors in the authentication process to enrich the existing system and yield strong authentication.

### *A. Two-factor based authentication*

Two-factor authentication (2FA) typically refers to using two means of identification to authenticate users. These could be a password together with a hardware token such as smart card, personal mobile device, or a hand-held token [9]. A common example of two factor authentication is when you use your ATM card to withdraw money from your bank account. Several two-factor authentication schemes have been proposed in the literature [8] [10] [11]. They were introduced to improve the shortcomings of password-based authentication and have enhanced security in authentication systems by providing an additional security layer (the token used). Indeed, smart cards have embedded electronic certificates that are used to identify the holder [9]. A comparative usability study of two-factor authentication was conducted in [12] and it was found that they are overall perceived as usable, regardless of motivation and/or context of use. Likewise several commercial two factor authentication systems exist today such as BestBuy's BesToken, RSA's SecurID, and Secure Computing's Safeword [14].

However, all 2FA schemes present drawbacks. Just as indicated in [8], research carried out in [13] on some two-factor authentication schemes pointed out they were vulnerable to impersonation attacks. This is due to the fact that users' credentials may be compromised and possibly abused if they lose their tokens. Also it is difficult to ensure that the person using the token and password to access the system is actually the owner who should be the legitimate person. A further drawback is the potential for the system to be unavailable to a valid user in situations where the token is lost or malfunctions [9]. Therefore, for a system whose users need high security requirements, two factor authentication schemes may be not secure enough [7] [8]. To solve this problem, three factor authentication schemes have been introduced.

### *B. Three factor based authentication*

This refers to using three means of identification to authenticate users such as a password together with a hardware token and the user's unique physical or behavioral characteristics (referred to as biometrics). The latter can include fingerprints, voiceprints, hand geometry, retinal or iris scans, handwriting, or keystroke analysis [2] [6] [9].

### *C. Biometric authentication*

Biometric characteristics are believed to be a reliable authentication factor since they provide a potential source of high-entropy information and cannot be easily lost or forgotten. Despite these merits, biometric authentication has some imperfect features. Unlike password, biometric characteristics cannot be easily changed or revoked. Some biometric characteristics (e.g., finger-print) can be easily obtained without the awareness of the owner. This motivates the three-factor authentication, which incorporates the advantages of the authentication based on password, smart-card and biometrics [5].

### *D. Zero Knowledge Proof authentications*

Zero Knowledge Proof (ZKP) is an interactive protocol which enables one party (the prover) to demonstrate knowledge of a certain secret to another party (the verifier) without revealing any information about the secret itself. This protocol can be solved by using mathematical problems like random numbers, discrete logarithms and integer factorization [3] [15] etc., to improve security. Typical ZKPs are based on several challenges and responses, involving a successive exchange of messages, which implies a very high communication cost. An extra consequence is the need to have a stable and continuous connection between nodes [16]. A ZKP must fulfill three main properties: completeness, soundness and zero-knowledge [3] [8] [17]. *Completeness* means that for any valid input, a prover P can always complete the proof successfully (i.e. the verifier accepts the prover's claim). *Soundness* ensures that no malicious prover P can construct a valid proof system (i.e. the verifier can never been convinced by any prover if its claim is false). *Zero-knowledge* guarantees that no malicious verifier V is able to derive extra knowledge from the interaction (i.e. the verifier cannot learn anything except the fact).

The concept of Non-Interactive ZKP (NIZKP) [8] [17] [18] [19] had been introduced to address the issue of successive exchange of messages in user authentication which has a non-negligible impact on the lifetime of the system in terms of resources usage. In an NIZKP, all of the challenges of a typical ZKP are condensed into a single package and sent in a single message [17]. This results in considerably reducing the time necessary to exchange messages, given that only a single message is sufficient to verify user's identity. In this research, node authentication is performed via a NIZKP-based approach using digital signature called DS-NIZKP. This approach employs digital signature with hash function to provide zero-knowledge and authentication of both the sender and message.


https://sites.google.com/site/ijcsis/
ISSN 1947-5500

*E. Cryptography and Digital Signature*

Cryptography is the science and study of secret writing whereby sensitive information can be stored or transmitted across insecure networks (like the Internet) so that it cannot be read by anyone except the intended recipient. The foundations of cryptography are the paradigms, approaches, and techniques used to conceptualize, define, and provide solutions to natural security concerns (such as confidentiality, data integrity, access control, and authentication) [21]. Figure 1 shows how cryptography works. A cryptographic algorithm works in combination with a key (a word, number, or phrase) to encrypt the plaintext.

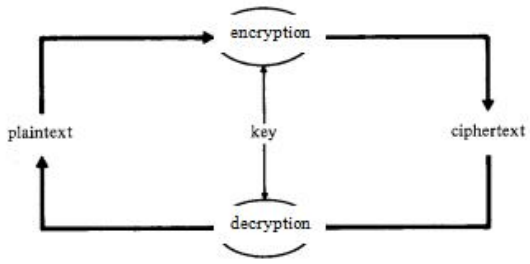


*Figure 1: How cryptography works*

The methods of cryptography render a message unintelligible to outsiders by various transformations of the text [20]. Plaintext is data that can be read and understood without any special measures. The method of disguising plaintext in such a way as to hide its substance is called encryption. Encrypting plaintext results in unreadable text called ciphertext. Encryption to ensure that information is hidden from anyone for whom it is not intended, even those who can see the encrypted data. The process of reverting ciphertext to its original plaintext is known as decryption.

There are two kinds of cryptography: conventional and modern cryptography. In conventional cryptography (also called secret-key or symmetric-key encryption), one key is used both for encryption and decryption whereas in modern cryptography (also called public-key or asymmetric-key encryption) two different keys are used. For better understanding, we present a brief review of the cryptographic concepts that are relevant to the construction of our authentication scheme.

**Digital signature** is one of the first tasks that joined encryption to form modern cryptography. A digital signature is an authentication mechanism that enables the creator of a message to attach a code that acts as a signature. Typically the signature is formed by taking the hash of the message and encrypting the message with the creator's private key [20]. DS ensures the source, integrity and non-repudiation of the message. Digital signatures employ asymmetric encryption and typically consist of three algorithms:

- A *key generation* algorithm: generates *private key* and *public key* for a signer
- A *signing* algorithm: produces a signature from a given a message and a private key
- A *signature verifying* algorithm: given the message, public key and signature, either accepts or rejects the message's claim to authenticity.

Since the advent of the digital signature concept, many signature schemes have been proposed [8] [20]: RSA, ElGamal, Shnorr, DSS. All these DS schemes have the following requirements [20]:

- A signature must be a bit pattern that depends on the message being signed,
- A signature must use some information unique to the sender to prevent both forgery and denial,
- It must be relatively easy to produce the digital signature,
- It must be relatively easy to recognize and verify the digital signature,
- It must be computationally infeasible to forge a digital signature, either by constructing a new message for an existing digital signature or by constructing a fraudulent digital signature for a given message,
- It must be practical to retain a copy of the digital signature in storage.

A secure hash function, embedded in a scheme, provides a basis for satisfying these requirements. However, care must be taken in the design of the details of the scheme.

A **hash function** is a one that takes a variable-length message as input and transforms it into a fixed-length output called hash value or message digest. The main goal of a hash function is data integrity. Indeed, a change to any bit in the message, results with high probability, in a change in the hash value. Hash functions are employed for a wide range of applications such as Message Authentication and Digital signatures. The general operation of a cryptographic hash function is illustrated on Figure 2.

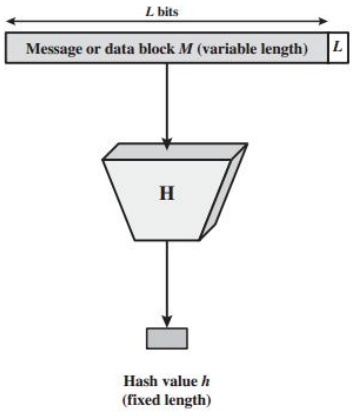


*Figure 2: Block diagram of a hash function, source [20]*

Here we consider cryptographic hash function which meets three main properties [8] namely:

- The hash value is easy to compute for any given input message


https://sites.google.com/site/ijcsis/
ISSN 1947-5500

- It is unfeasible to find two distinct messages with the same hash value
- It is unfeasible to recover a message from it hash value

A good hash function has the property that the results of applying the function to a large set of inputs will produce outputs that are evenly distributed and apparently random [20]. The Secure Hash Algorithms (SHA) are a family of cryptographic hash functions which are commonly used. SHA-3 is the latest of this family and is based on an algorithm called Keccak. Keccak is a permutation based algorithm that NIST selected as the winner of the SHA-3 Cryptographic Hash Algorithm Competition [23]. Even though it supports the same hash lengths as SHA-2, SHA-3 has the ability to run on many types of devices, as well as its facility in performing better than the others. With Keccak, it is possible to target a given security strength level by choosing the appropriate capacity, i.e., for a given capacity c, Keccak is claimed to resist any attack up to complexity $2^{c/2}$ [17] [23].

In this work, we use a general method for constructing signature out of length-restricted ones. The method consists of hashing a message/document into a short (fixed-length) string (using a SHA-3 algorithm), and applying the RSA signature scheme to the resulting hash-value. With the RSA approach, the message to be signed is input to a hash function that produces a secure hash value of fixed length. This hash value is then encrypted using the sender's private key to form the signature. Both the message and the signature are then transmitted. The recipient takes the message and produces a hash value. The recipient also decrypts the signature using the sender's public key. If the calculated hash code matches the decrypted signature, the signature is accepted as valid. Because only the sender knows the private key, only the sender could have produced a valid signature.

## III. PROPOSED APPROACH

In this section, we present the proposed DS-NIZKP-based authentication scheme. We first discuss the system setup and give an overview of the protocol. Then we elaborate on the details in each phase of the authentication.

### *A. Preliminaries*

The main goal of this paper is to propose a strong authentication scheme based on ZKP to avoid disclosure of sensitive information during communication and communication cost is least. Strong authentication involves two or more factors when authenticating users. The proposed scheme uses a three-factor authentication even though it can be adapted to two or more factors. This means it can allow the server to decide the authentication factors in a user's authentication (instead of all three authentication factors). We therefore need a dynamic data structure (such as a linked list) to store users identity information. Since digital signature employs asymmetric cryptography, we assume that an appropriate key distribution and management system is put in place.

Last, in the proposed scheme clients' authentication information (password, smart card number and biometric information) are kept secret from servers. This means that these information are not stored in clear on the server, rather they are disguised. This not only protects user privacy but also prevents a single-point failure (e.g., a breached server) from undermining the authentication level of other services. Furthermore, the verification of all authentication factors is performed by the server.

We consider authentication to be a function of three (03) parameters: **K, P** and **B**, where K represents something users know, P something users possess and B a biometric characteristic of the user. What the user knows yields id1; what he has yields id2 and finally what he is yields id3. The identity of the user is then composed of id1, id2 and id3, which is stored in the data structure. Each component of this identity is then hashed, using a hash function, to obtain a digest of fixed size. It is worth recalling that hash functions have the following properties:

- One way; which means they cannot be reversed
- The output (digest or hash value) does not reveal information on input
- Collision resistance; which means that it's hard to find collisions (i.e. different messages with same hash)

### *B. DS-NIZKP-based authentication*

The proposed solution consists of 03 stages as described below. A user requesting access to the system needs to go through these various stages during its authentication.

**Stage 1: Building user's identity (ID)**

We consider authentication to be a function of three (03) parameters: K, P and B

$$\textbf{Auth} = \textbf{F}\ (\textbf{K}(), \textbf{P}(), \textbf{B}())$$

*with **K** being something users know, **P** something users possess and **B** a biometric information of the user*

What the user knows yields **id1**; what he has yields **id2** and finally what he is yields **id3**. The identity of the user is then composed of id1, id2 and id3 as on shown on Figure 2, which is stored in the data structure (of type linked list).

**ID = id1; id2; id3**

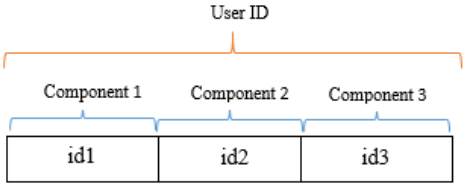


*Figure 3 : Components of a user's ID*


https://sites.google.com/site/ijcsis/
ISSN 1947-5500

Each component of this identity is then hashed, using a hash function, to obtain a digest of fixed size. Hashing prevents from revealing personal information of the user, therefore satisfying the zero knowledge property of zero knowledge protocols.

**Stage 2: Generating user's signature (Sign)**
The user's signature is obtained by encrypting each hashed component of the identity above with the user's private key. This encryption provides confidentiality, non-repudiation and integrity.

$$\mathbf{Sign = E_{PR}\ (ID)}$$
$$\mathbf{= E_{PR}\ (id1)\ E_{PR}\ (id2)\ E_{PR}\ (id3)}$$

*where **E** is an encryption function and **PR** the user's private key*

After the signature has been generated, the pair user identity and signature, (ID, sign) is sent to the server for verification.

**Stage 3: Verification of user's information**
Upon reception of the user's authentication information, the server proceeds to it verification as follows:

- **Verification of user's signature**: the server decrypts Sign using the user's public key and verifies with the received ID information. This is done by comparing each component of the obtained signature, Sign', with those of the ID received.

  **Sign' = DPU (Sign)**

  **= DPU (EPR(id1) EPR (id2) EPR (id3))** *where D is a decryption function and PU the user's public key*

  **Cmp (Sign, Sign') = bool** *where Cmp is a comparison function and bool a Boolean value*

  This assures authentication, non-repudiation and integrity because only the claimed user could have sent such information since he is the only one possessing his private key which was used to encrypt. If all comparisons are successful, then user's signature is valid and its authentication information are reliable. The server can now proceed to the verification of its identity.

- **Verification of user's identity**: the server compares the received information on the user's ID (id1, id2 and id3) with those stored on the server. Again, this is done by comparing each component of the ID with that on the server. If all comparisons are successful, then the user is accepted and access to the system is being granted.

## IV. SECURITY ANALYSIS AND PERFORMANCE EVALUATION

This section assesses and provides a formal security analysis of the proposed DS-NIZKP scheme. We start by briefly describing the completeness, soundness and zero-knowledgeness of our approach. Then we analyze and provide some security proofs in relation with the proposal.

**Completeness**: this property guarantees that both the prover and the verifier will follow the normal protocol. Given that users authentication information are stored on the server, if both parties (in this case the user and the authentication server) follow the protocol properly, the valid identity of a user requesting access to the system can be accepted by the server with probability one.

**Soundness**: this property ensures that no malicious prover can construct a valid proof. Considering that the authentication server is honest (follows the protocol), then with overwhelming probability, an unauthorized user will not be able to cheat by sending an invalid identity since it is composed of several components and signed using user's private key. An overwhelming probability means the probability is 1 - a negligible probability [22].

As described in section 4, the user generates, hashes and signs its identity to the authentication server. The server checks the validity of the user by checking its signature. For the user to be valid, each component of the obtained signature Sign' should match with those of the one received Sign. Once the user's validity is confirmed, its identity can now be checked. The validity of the user identity ID clearly depends on the validity of its signature Sign and the validity of each of its components id1, id2, and id3. Thus, if the server is honest, it will be able to easily detect the invalid identity (in case there is any invalid ID component by a user). So, there is no way for an unauthorized user to succeed with invalid ID.

**Zero-knowledge**: this property refers that no information whatsoever except the validity of the prover's claim flows to the verifier. On the one hand, users information are not stored in clear on the authentication server. They can be disguised using a hash function such that the server itself cannot learn anything about the user's identity. This also prevent from attacking the server and stealing sensitive information on the users. On the other hand, DS-NIZKP uses a cryptographic hash function to disguise user's personal information before sending to the server. Such functions have the property of making it unfeasible to recover a message from it hash value (one way functions; which means they cannot be reversed). This means, the authentication server receive only the necessary values from the user to individually prove the validity of its identity, but the user does not send any value to the server such that the server individually can reveal the value of a identity. Thus zero-knowledgeness property of the protocol is fulfilled.

From the above discussion we see that, the DS-NIZKP scheme for user authentication presented in this paper satisfies completeness, soundness, and zero-knowledgeness properties of a ZKP. This represents the first security measure for our approach.


https://sites.google.com/site/ijcsis/
ISSN 1947-5500

The security of the proposed scheme also depends on standards used to implement it: ZKP, digital signature hashing and encryption. The SHA-3 family algorithms which we have chosen are designed to provide special properties, such as resistance to collision, pre-image, and second pre-image attacks [23]. These hash functions are also components for many important information security applications, including the generation and verification of digital signatures [23] which have been used to implement the ZKP. Therefore, the proposal does not suffer from usual attacks based on cryptographic operations (also like identity theft and Man in the Middle attack), because its security is supported by NIZKP based on digital signature, current standard hashing and encryption. It is worth indicating that the type of encryption used here is homomorphic encryption. Homomorphic encryption is a form of encryption that allows computation on ciphertexts, generating an encrypted result which, when decrypted, matches the result of the operations as if they had been performed on the plaintext. The purpose of homomorphic encryption is to allow computation on encrypted data.

## V. CONCLUSION AND FUTURE WORKS

In this paper, a flexible and lightweight strong authentication scheme (DS-NIZKP) for distributed systems has been presented. DS-NIZKP uses three factors to authenticate users and a dynamic data structure to store user's identity components, enabling the approach to be used with less or more authentication factors. It is designed based on non-interactive ZKP and homomorphic encryption by allowing each user to self-generate a number of authenticated identities to prove his or her legal status when communicating with peer users, service providers, or other infrastructure. To implement the ZKP concept, digital signatures have been employed for the protection of user privacy. The proposed approach is simpler in that, contrary to typical ZKP systems, there is no interaction between the prover and the verifier: just a single message is necessary.

In the short term, we plan to use and apply the proposed model in other related issues. Also, we expect to propose a distributed approach for the proposed solution, where there will be an agent responsible for authenticating each factor with the server.

## REFERENCES

[1] Thomas Y .C. Woo and Simon S. Lam, Authentication for Distributed Systems, In Internet Besieged: Countering Cyberspace Scofflaws. 1997. ACM Press and Addison-Wesley.

[2] Kristian Skračić, Predrag Pale and Branko Jeren, A distributed authentication architecture and protocol, Tehnički vjesnik 24, Suppl. 2 2017, pp. 303-311

[3] Gerardo I. Simar, A Primer on Zero Knowledge Protocols, Universidad Nacional del Sur, 2002

[4] J. Wei, L. Dan, L. Feng and B. Elisa, "Randomized and Efficient Authentication in Mobile Environments", Cyber Center Publications, Paper 633, 2014

[5] X. Huang, Y. Xiang, A. Chonka, J. Zhou, and Robert H. Deng, A Generic Framework for Three-Factor Authentication: Preserving Security and Privacy in Distributed Systems. IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, 2010

[6] Jiangshan Yu, Guilin Wang, Yi Mu and Wei Gao, An Efficient Generic Framework for Three-Factor Authentication with Provably Secure Instantiation, IEEE Transactions on Information Forensics and Security, Volume 9 Issue 12, pp 2302-2313, December 2014

[7] J. Ngozi Oruh, Three-Factor Authentication for Automated Teller Machine System, IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS), ISSN: 2249-9555 Vol. 4, No.6, December 2014

[8] J. Yu, Remote user authentication in distributed networks and systems, Master's thesis, University of Wollongong. 2012

[9] Thomas G. Calderon and Colin G. Onita, Big Data and the Perceived Expectations Gap in Digital Authentication Processes, Journal of Forensic & Investigative Accounting, Volume 9: Issue 2, July–December, 2017

[10] M. Singhal and S. Tapasw, Software Tokens Based Two Factor Authentication Scheme, International Journal of Information and Electronics Engineering, Vol. 2, No. 3, May 2012

[11] S. Yoo, Seung-jung Shin and Dae-hyun Ryu, An Innovative Two Factor Authentication Method: The QRLogin System, International Journal of Security and Its Applications Vol. 7, No. 3, May, 2013

[12] E.De Cristofaro, H. Du, J. Freudiger and G. Norcie, A Comparative Usability Study of Two-Factor Authentication. 8th NDSS Symposium, January 2014

[13] G. Wang and F. Bao, Cryptanalysis of timestamp-based password authentication schemes using smart cards, In Proceedings of the 8th international conference on Information and Communications Security, ICIC'06, pages 399-409, Springer-Verlag, 2006

[14] A. Nath and T. Mondal, Issues and Challenges in Two Factor Authentication Algorithms, International Journal of Latest Trends in Engineering and Technology (IJLTET), ResearchGate, 2016

[15] S. Kumar Mandal and A. R Deepti, A General Approach of Authentication Scheme and its Comparative Study, International Journal of Computer (IJC), 2017

[16] U. Feige, A. Fiat, A.Shamir, Zero-knowledge proofs of identity. J. Cryptol. 1988, 1, pp 77–94

[17] F. Martín-Fernández, P. Caballero-Gil and C. Caballero-Gil, Authentication Based on Non-Interactive Zero-Knowledge Proofs for the Internet of Things, MDPI Journal Sensors 2016, 16, 75, doi:10.3390/s16010075

[18] M. Blum, P. Feldman, S. Micali, Non-interactive zero-knowledge and its applications, ACM Symp. Theory Comput. 1988, doi:10.1145/62212.62222.

[19] C. Rackoff and D.R. Simon, Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack, In Advances in Cryptology, Feigenbaum, J., Ed.; Springer: Berlin, Gernamy, 1992; Volume 576, pp. 433–444

[20] W. Stallings. Cryptography and Network Security Principles and Practice, 5th Edition, 2006 Pearson Education, Inc., publishing as Prentice Hall

[21] O. Goldreich, Foundations of Cryptography Volume 2, Basic Applications, Cambridge University Press, 2004

[22] K.R. Iversen, The Application of Cryptographic Zero-Knowledge Techniques in Computerized Secret Ballot Election Schemes, Ph.D. dissertation, IDT-report, 1991:3, Norweigan Institute of Technology, February, (1991)

[23] Federal Information Processing Standards Publication, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, MD 20899-8900 May 2014