A New Mutuel Kerberos Authentication Protocol for Distributed Systems

Zakariae Tbatou, Ahmed Asimi, Younes Asimi, Yassine Sadqi, Azidine Guezzaz
(Corresponding author: Zakariae Tbatou)

Department of Mathematics, Faculty of Sciences, Ibn Zohr University, Agadir, Morocco
B.P 8106, City Dakhla, Agadir, Morocco
(Email: tbatou.zakariae@gmail.com)
(Received July 18, 2016; revised and accepted Sept. 25 & Oct. 25, 2016)

Abstract

In recent years, distributed systems, including cloud computing, are becoming increasingly popular. They are based on traditional security mechanisms that focus on access control policies and the use of cryptographic primitives. However, these mechanisms do not implement some more advanced security properties, including authentication policies. Kerberos V5, the most recent version, is a successful protocol that is designed to authenticate clients to multiple networked services. In this paper we propose a new mutuel Kerberos authentication protocol for distributed systems based upon Kerberos V5 and Diffie Hellman models. it is composed of three phases: 1) registration phase, based on the Diffie Hellman model, enabling the design and reliable exchange of client's authentication parameters to the authentication server side; 2) communication phase, based upon the two functions S2KexS () and DKexS (), which aims to the exchange of encryption keys and creates a secure the communication channel between client and server of services and 3) renewal phase for updating the client authentication parameters. Our security analysis and performance evaluation demonstrate that our scheme creates a secure channel to a more secure password exchange. Hence, it reduces the chance that a password will be guessed from the parameters stored or exchanged between client and authentication server, which make our proposed protocol efficient against dictionary and brute force attacks. The results proved by the behavior study show the success of our scheme and the easily of implementation. *Keywords: Authentication; Cloud Computing; Cryptographic Primitives; Diffie Hellman Model; Distributed Systems; Kerberos V5*

1 Introduction and Notations

Most authentication mechanisms are based only on password [8, 11, 20, 26]. In these regimes, the distant server maintains a table to record information of each user's password, and exploits them to verify the corresponding user privileges. However, although they are widely used in many applications in real life, authentication systems based on password suffer from several attacks [1, 3, 20, 27], such as dictionary attacks [23] brute force, steals data, Guessing Attacks [4, 14, 22, 27], etc. to respond to these issues, Kerberos V5 presents a strong protocol of network authentication for client/ server applications [13, 31]. It uses a (KDC), and tickets distribution center (TDC) [12, 24, 25]; in the sense that it never transmits passwords [21, 25, 31]. It exchanges encrypted messages with limited life by adding entities called tickets [12, 25]. All authentication requests are routed through the centralized KDC server [24, 25]. The latter defines a unique namespace for different clients [12]. In our approach, we assume that the communication between realms and service servers is based on Single Sign On.

In this paper, we begin by presenting the authentication dialog and the different cryptographic primitives for keys generation. In the third section, we propose a new communication scheme with a description of the three phases: 1)the registration phase based on Diffie Hellman model [7] and a dynamic salt generator (RGSCS) [2]); 2)the communication phase based upon the two functions S2KexS () and DKexS ().

The first function aims to generate a basic key from the footprint of the password and a dynamic salt per session. Based on this basic key, the second function is designed to generate encryption keys and 3)the renewal phase for updating the clients authentication parameters. The fourth section describes a behavioral study of the three phases with the use of regenerator of salts (RGSCS), dynamic and cryptographically secure. The five section studies the security analysis of our approach by evaluating the impact of its three phases on the robustness of the Kerberos V5 protocol. We end this paper with a conclusion. In all that follows, we denote by Table 1.

Table 1: Notations

Symbols	Meaning
C:	Client.
S:	Server of services.
KDC:	Key Distribution Center.
ID:	Identity of client.
ID_R:	Identity of releam.
Pwd:	User's password.
$\mathbb{N}$:	The set of natural numbers.
$salt_i$:	Dynamic pseudorandom sequence.
$\|\|$:	Concatenation.
==:	Comparaison.
mod:	Modulo operation.
$K_{x,y}$:	Session key shared between x and y.
$\{m\}K_x$:	m encrypted by the secret key.
$T_{x,y}$:	Ticket of x to use y.
$A_{x,y}$:	Authenticator of x for y.
$\mathbb{F}_p$:	Finite field of order a prime number p.
$\mathbb{F}_p^*$:	Cyclic multiplicative group of all non zero elements in $\mathbb{F}_p$ of order $p-1$.
$S(i)$:	$(i+1)^{th}$ binary string position of S.

2 Related Work

The modern Kerberos has undergone several major revisions. In each review, significant improvements have been made like scalability and security. The version 1 through 3 were used internally and as to version 4 was the first version distributed to the public was Kerberos V4, which has been limited in some nations due to the limitations of used encryption algorithms. These limitations made norms to evolve a new protocol that contains all the features presented in the Kerberos V4, with the addition of features such as extensible encryption types and more transparent authentication to create the version 5 of Keberos [13, 25]. After all these changes and with the development of computer system, Kerberos V5 still vulnerable against attacks such as attacks by brute force and dictionary. They still represent a real challenge for this protocol. These conclusions made thinking several researchers to propose solutions such as the use of asymmetric cryptographic primitives [17], in order to make the keys generation more reliable, or the introducing of new technologies such as smart card [16]. In this section, we present the communication phase based on two strong points: cryptographic primitives and tickets, and the various requests exchanged between a client and the KDC server to access a service.

2.1 Communication Dialogue

The communication dialogue in Kerberos V5 introduces three entities: a client, a centralized KDC server and a server of services. Authentication requests are routed through the centralized server KDC [12, 25] as described in Figure 1.

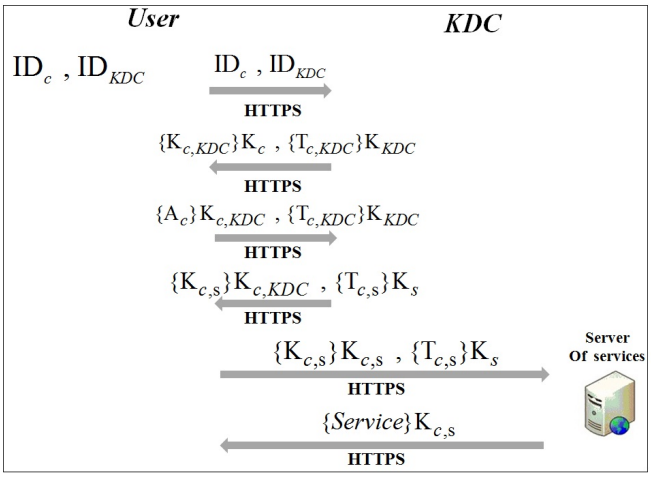

Figure 1: Description of Kerberos V5 queries

In Kerberos V5, the ticket distribution center acts as an intermediary of various requests exchanged between client and server of services to authenticate the client before access to the wanted service, based on two entities: tickets, which are used to authenticate client to the ticket distribution center and an authenticator to validate the client's identity to the server of services.

2.2 Cryptographic Primitives and Diffie Hellman Problem

Kerberos V5, in its communication phase, uses three encryption keys. Referring to [12, 25], the steps to generate these three keys are as follows:

- Regeneration of the basic key either by the random-to-key () function from a random bit string, or by the String-to-key() function from a password and a salt.
- Regeneration of these three keys associated to this based key by the key derivation function called Derived-key().

The Diffie-Hellman protocol is a method for two computer users to generate a shared private key with which they can then exchange information across an insecure channel. We refer to [15, 19] and we deduce the following results.

Definition 1. *A primitive element of $\mathbb{F}_p$ is a generator of a cyclic units group $\mathbb{F}_p^*$.*

Definition 2. *The Diffie Helman problem is the following : given a prime number, a primitive element g of $\mathbb{F}_p$, and $g^a \ mod \ p$ and $g^b \ mod \ p$, find $g^{ab} \ mod \ p$.*

Definition 3. *The generated Diffie Helman problem is the following : given a finite cyclic group G, a primitive element g of G, and group elements g^a and g^b, find g^{ab}.*

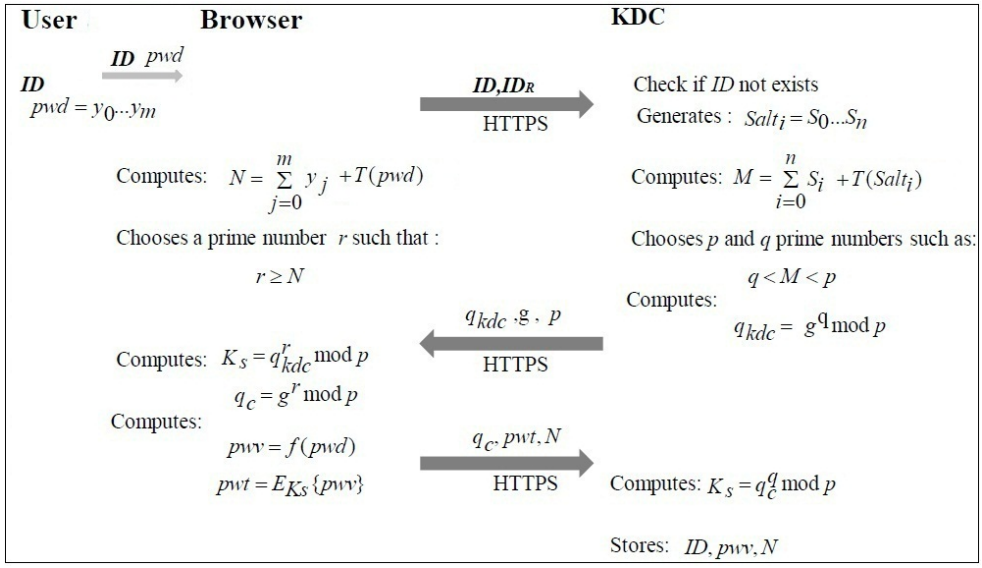

Figure 2: Description of the new registration phase

According to Definitionss [1, 2, 3] the use of Diffie-Hellman causes some problems at the implementation level: 1) the problem to determine with effective way the primitive elements of a finite field [5, 6]; 2) the difficulty of implementation specifically the complexity of the computation time and performance especially in systems require the notion of time [10], and 3) the synchronization problem relatively to the time system. In our approach, we have took into consideration these problems with using the Diffie-hellman principle by the choice of a finite field $\mathbb{F}_p$ with $p = 2^n + 1$ and its primitive elements which are the form 3^{2m+1} modulo p for all $m \in \mathbb{N}$.

3 Description of Our Approach

The scheme of our conception consists of three entities: 1) Kerberos client that belongs to the KDC realm; 2) Browser that supports HTTPS for a more secure data exchange and cryptographic primitives, virtualisation functions and hash functions; 3) KDC server, which is the key distribution center, provides symmetric cryptographic primitives, virtualization functions and hash functions. It is composed of a basic three storing identification parameters assigned to each user identified by ID. These parameters are used to authenticate users during the communication phase and can be easily changed in the renewal phase, and which are successively rated:

ID	*pwv*	*N*

- *ID*: User identification.
- *pwv*: Footprint of the password. In our proposal, it will be used for generating keys encryption / decryption to ensure:
  1) The user identification during communication and renewal phases.
  2) The confidentiality of messages exchanged between users and the KDC server.
  3) The confidentiality of the new password chosen by users in the renewal phase.
- *N*: Integer number regenerated from the password.

3.1 Conception of Our Approach

Our authentication scheme is based on three phases: registration, communication and renewal phases.

3.1.1 Registration Phase

This phase, regenerates its own authentication settings using a username and password of user not shared between the browser and the KDC, as described in Figure 2.

In this process, the KDC generates for each user three authentication parameters, based on a salt generator which generates different salts for each user. At the client side, each client must have a valid password and a unique ID that does not exist in the database. The dialogue of the registration phase is described as follows:

- The client sends its ID and ID_R of releam Which he wants to register to the KDC.
- The KDC server checks the existence of the *ID*.
  - If it exists, it returns an error message.
  - Otherwise, it
    - Generates a first *salti*.
    - Calculates M that is equal to the sum of the bits of *salti* and *salti* length.
    - Chooses two prime numbers p and q with p upper than M and q lower than M.

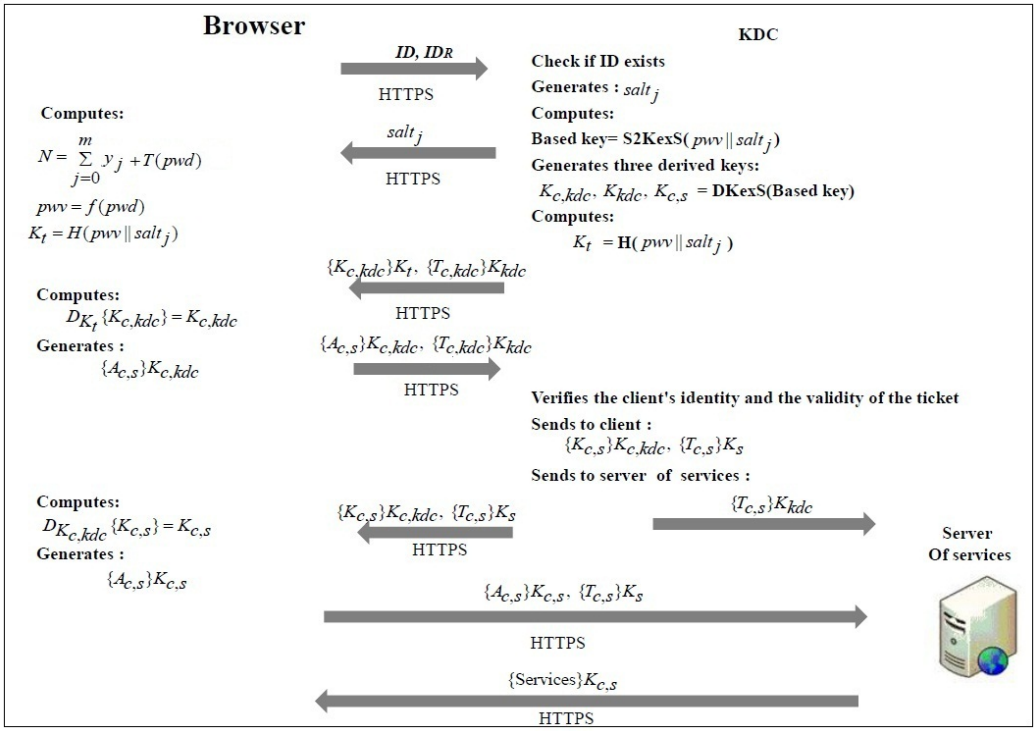

Figure 3: Description of the new communication phase

* Chooses a number g in order that g is a divisor of M.
* Calculates $q_{kdc} = g^q \bmod p$
* Sends q_{kdc}, g and q to the client.

- The client:
  - Calculates N, which is equal to the sum of the password bits and the password length.
  - Chooses a prime number r upper than N.
  - Calculates $q_c = g^r \bmod p$.
  - Calculates $pwv = f(pwd)$ where f is a virtualization function.
  - Calculates the key $K_s = q_{kdc}^r \bmod p$.
  - Sends q_c, $\{pwv, N\}K_s$ to KDC .
- The KDC server:
  - Calculates the key $K_s = q_c^q \bmod p$.
  - Decrypts $\{pwv, N\}K_s$ and obtains pwv, and N.
  - Stores ID, pwv and N.

3.1.2 Authentication and Identification Phase (Communication)

In this phase each user must prove his identity (ID) to the KDC server, specifically the KDC that must authenticate the user because the Kerberos system is based on a trusted third party [8, 20, 24, 25]. For this reason the client sends his ID and the ID_R of its realm (authentication without sending the password [12, 21]) to the KDC server. it last checks the ID in the database, if it exists, the KDC generates a basic key from client authentication parameters stored in the database if not it returns a message error. The key generation has been enhanced by new features $S2KexS$ and $DKexS$ [30] to make the generation key dynamic.

The dialogue of the communication phase is described as follows (see Figure 3):

- Client :
  - Sends his ID and the ID_R of its releam to KDC server.
  - Calculates N.
  - Enters the password pwd and calculates $pwv = f(pwd)$.
- The KDC server:
  - Verifies his own ID_R and checks the existence of user ID, if doesn't exist the KDC sends an error message, otherwise.
  - Calculates a based key with the function $S2KexS$ from pwv stored in the database and a new regenerated $salt_i$
  - Calculates three derived keys $K_{c,kdc}$ K_{kdc} and $K_{c,s}$ with the key derivation function from the based key $DKexS(besed\ key)$.
  - Calculates a temporary key $K_t = H(pwv||salt_i)$.
  - Encrypts $K_{c,kdc}$ with K_t.

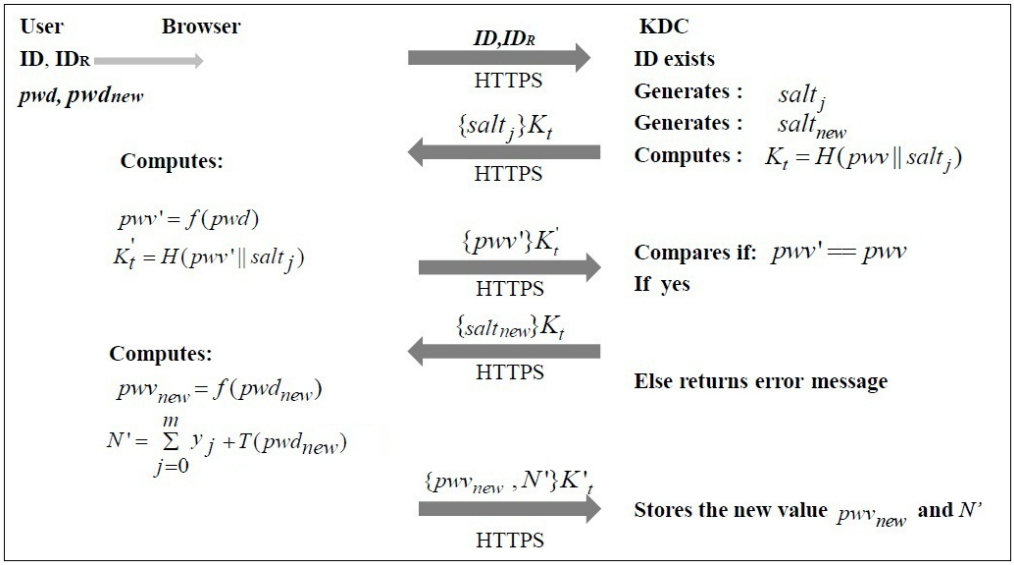

Figure 4: Description of the renewal phase

- Encrypts $T_{c,kdc}$ with K_{kdc}.
- Sends $\{T_{c,kdc}\}K_{kdc}$, $\{K_{c,kdc}\}K_t$ and $salt_i$ to the client.

- The client:
  - Calculates $K_t = H(pwv||salt_i)$.
  - Finds the $K_{c,kdc}$.
  - Generates an authenticator $A_{c,kdc}$ which contains the requested service, the calculated number N and others authentication parameters.
  - Sends $\{A_{c,kdc}\}K_{c,kdc}$ and $\{T_{c,kdc}\}K_{kdc}$ to KDC.
- The KDC server:
  - Finds $A_{c,kdc}$ and $T_{c,kdc}$.
  - Checks the validity of the ticket time and the client's identity from $A_{c,kdc}$ parameters.
  - Encrypts $K_{c,s}$ with $K_{c,kdc}$.
  - Creates a ticket $T_{c,s}$ that will be shared between the client and the server of services.
  - Encrypts $T_{c,s}$ with K_s.
  - Sends $\{T_{c,s}\}K_s$ and $\{K_{c,s}\}K_{c,kdc}$ to the client.
- Client:
  - Decrypts $\{K_{c,s}\}K_{c,kdc}$ and gets $K_{c,s}$.
  - Generates $A_{c,s}$ which contains the service and client authentication parameters and encrypts it with $K_{c,s}$.
  - Sends $\{A_{c,s}\}K_{c,s}$ and $\{T_{c,s}\}K_s$ to server of services.
- Server of services:
  - Decrypts $\{T_{c,s}\}K_s$ with K_s and checks the client's identity and the validity of the ticket time..
  - If the identification is successful, it encrypts the requested service with the key $K_{c,s}$ and sends the message to the client. Otherwise the server of services sends an error message.

3.1.3 Renewal Phase

This phase allows the renewal of client authentication parameters. It represents the most important phase especially for new users, because it enables the exchange of the new parameters in an environment more secure than the registration phase. In this phase, we must ensure the identity of the user, mutual authentication and validity of the new password as described in Figure 4.

In this phase, it should be noted that the client is already logged into his session so the encryption keys are already shared. So the client must enter his old password to validate the authentication parameters with the KDC server, then he enters his new password.

- Client sends his ID and ID_R of releam to the KDC.
- The KDC server:
  - Verifies his ID_R and checks the existence of ID. if doesn't exist, it returns an error message, otherwise:
  - Generates two new salts $salt_{new}$ and $salt_j$.
  - Calculates K_t which is equal to hashed pwv concatenated with $salt_j$.
  - Sends $salt_j$ to the client.
- Client:
  - Enters his pwd.

Figure 5: Behavioral study of the registration phase for three iterations

  - Calculates $pwv' = f(pwd)$ and $K_t = H(pwv'||salt_j)$.
  - Sends $\{pwv'\}K_t$ to the KDC.

- The KDC server:
  - Decrypts $\{pwv'\}K_t$ and compares pwv' with pwv:
  - If $pwv' == pwv$ then $K_t == K'_t$ therefore the server sends the $salt_i$ encrypted with K_t to the client. Otherwise it sends an error message asking him to send his ID.

- Client:
  - Gets the $salt_{new}$ using K_t.
  - Calculates the new value of N.
  - Calculates $pwv_{new} = f(pwd_{new})$.
  - Sends $\{pwv_{new}, N'\}K_t$ to KDC.

- Server KDC:
  - Decrypts $\{pwv_{new}, N'\}K_t$ with K_t.
  - Updates the server database with new values of pwv_{new} and N'.

4 Behavior Study

After presenting the purpose of the integrated regenerator RGSCS [2] in different phases, in this section, we focus on behavioral study of registration and communication phases to test the influence of RGSCS on our proposal. We begin then by studying the impact of salts regenerated by RGSCS for a low and redundant given password 'aaaaaa', on the generation of encryption keys in the registration phase, and generation of session key in the communication phase by studying the correlation of the generated binary sequences.

4.1 Behavioral Study of the Registration Phase

In the figure Figure 5, we have implemented the registration phase using PHP 5 to program the various functions; DES mode CBC as an encryption algorithm, a virtualization function based on the dynamic rotation [1], the RGSCS generator and the Diffie Helmman protocol based on a finite field $\mathbb{F}_p$ of characteristic $p = 2^n + 1$ [7] having as primitive elements the numbers 3^{2m+1} modulo p for all $m \in \mathbb{N}$. The hardware used in our experiments is a AMD $E-300$ CPU 1.3 GHz and 4Go as RAM running under Windows 7. In our case, we took the number 3 as a small primitive element of $\mathbb{F}_p$ to evaluate our results even this primitive element make our protocol dynamic

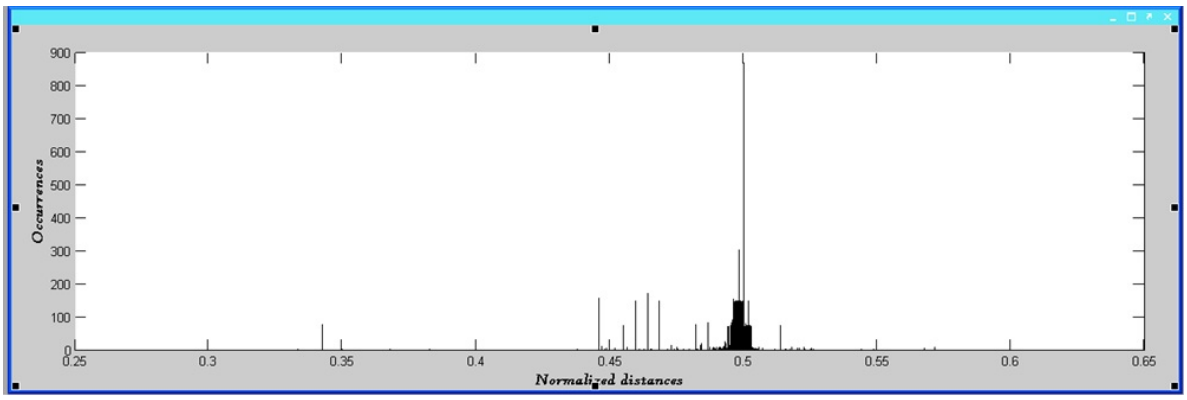

Figure 6: Study of the no correlation of the keys K_t for 200 sessions with the same password, and different salts

and per user.

For three iterations and a given password, we analyze the entities regenerated for the client and the KDC server and we deduce the following results:

- The sent messages are not related to the original password.
- The footprint of the password is unpredictable.
- The encryption keys are dynamic and per session.

4.2 Behavioral Study of the Communication Phase

In our approach the session keys are dynamic, per session and have a variable size. However, the behavioral study of these keys requires a normalized Hamming distance, named D, defined in [2] by:

$$D(S,S') = \frac{\sum_{i=0}^{k-1} ((S(i \ mod \ K) + S'(i \ mod \ K')) \ mod \ 2)}{k} \quad (1)$$

with S and S' are two binary strings having period successively K and K' not necessary the same and $k = lcm(K, K')$. This function D allows the estimation of correlation between binary sequences not necessary with the same length. Asimi et al [2] found that two binary strings S and S' are weakly correlated if $D \simeq 0.5$.

Propriety 1. *: Let S and S' be two periodic binary strings. we say that S and S' are weakly correlated if $D(S,S') \simeq 0.5$.*

In Figure 6, even under restricted cases the results are accumulated in the vicinity of 0.5, which means that the keys used and associated to the same password are not correlated. Therefore, knowledge of information on the key gives no information on the other. This is due to the uncorrelation of binary signals calculated by the hash function applied to the fingerprint password concatenated with a dynamic salt per session.

5 Security Analysis

The evolution of the computer system and the development of new technologies, the attacks become increasingly efficient. For these reasons, Kerberos has known several modifications to the levels of performance and functionality against these attacks. However Kerberos V5, the current version, with all its amelioration, was discussed by several security analysis [9, 13, 33, 31], those show its weaknesses specifically against the dictionary attack only in the communication phase.

In this section, we evaluate the security of our protocol by analyzing the level of influence using addition salt to the password, and the impact of the Diffie Hellman principle [7] against different types of attacks. Further, we discuss the impact of adding dynamic salt per session to the password in both client side and KDC server side. In the client side, the addition of a dynamic salt per session to the password, and the application of virtualization function make the authentication process by password unbreakable. They reduce the chance of password divination attacks such as brute force and dictionary attacks. In the other hand, storing the password footprint (dynamic password disturbed by salt in our case) is stronger than storing the clear password in KDC database server. This makes storage of password more reliable in the KDC server side.

5.1 Impact of Salt Upon Password

The majority of the applications users are conscious of authentication by passwords. It requires the storage of simple passwords in most cases [20, 26]. In parallel, other authentication alternatives have been proposed [34]. However, their use is too limited especially in web applications [32]. The description of Kerberos integrated a static salt (client address or the domain name) to disrupt the password used for the generation of encryption keys [12, 13].

This technique does not solve the problem of dictionary attack that represents a real challenge against the Kerberos authentication techniques [17, 31, 33]. To address this type of attack, our approach is based on the

Table 2: Comparison between our protocol and previous versions of Kerberos

Parameters	Previous version of Kerberos	Our protocol
Mutual authentication	OK	OK
Portability	OK	OK
Use of ticket	OK	OK
Use of expiration time	OK	OK
Use of Deffie Helman		OK
salt	static and per user	dynamic and per session
Session key	$K_s = H(pwd)$	$pwv = f(pwd)$ and $K_s = H(pwv\|\|salt_i)$
Based key	based on string-to-key function	based on S2KexS function
Derived key	based on derived key function	based on DKexS function
N		New authentication number calculated from password

RGSCS regenerator making the use of keys generation functions more robust, and who's their different outputs from a session to another.

As for the registration phase, the impact of salt used to disrupt the password makes it communication phase more reliable (registration of password footprint). Therefore the guess of original password either by listening to requests exchanged between the client and the KDC or by brute force is almost impossible.

5.2 Impact of the Diffie Hellman Principle

The principle of Diffie Hellman solved several types of attacks such as man in the middle [7]. It has undergone several changes [5, 6, 10] with the development of computers (computing speed, performance processors). The conjunction of this principle and the dynamic salt per session made the parameters used in our protocol more complicated and indefinable. This allows us to create a secure channel to a more secure password exchange. with this technology we have reduced the chance that a password will be guessed from the parameters stored or exchanged between client and KDC.

5.3 Robustness to the Dictionary Attack

Most password crackers are provided with standard dictionaries [23]. The experience allows that the Kerberos realm had already the strength of the password, reflecting authentication without sending it [12]. Although, the description of registration phase is not written in any reference, and the communication phase is based on a clearly stored password [18, 25, 28]. Our principle reduces the probability of finding the password is in the registration phase and communication phase. It is caused by disturbance by adding the dynamic salt per session and application virtualization function. Even if a hacker succeeded in capturing several messages, he will not have the opportunity to find the password in question by the dictionary attack.

6 Comparison Between Our Protocol and Previous Versions

Our protocol, which is a Kerberos V5 improvement, aims to ensure the confidential exchange between clients, authentication servers and services server. For these reasons our approach is based on tickets, the Diffie Hellman protocol and other functions namely: S2KexS function, DKexS function [30].

- Diffie Hellman algorithm allows confidential exchange of credentials authentication without requirement HTTPS.
- S2KexS function calculates a more robust and undevinable base key from a dynamic salt and a password digital print.
- DKexS function calculates three encryption keys used in the communication phase to ensure the confidentiality and integrity of data exchanged between clients, servers and services.

However the adding of pseudorandom regenerator, S2KexS function, DKexS function and Diffie Hellman protocol makes our protocol more robust. The comparison between our approach and the traditional Kerberos defined in [25] is described as follows:

7 Conclusion

Several extended authentication protocols have been described for strong password authentication [20, 27, 29]. For Kerberos, several solutions have been proposed such as using the smart card [16] or public keys [17] etc, but these techniques do not reduce the chance that the password is guessed and the rest of the protocol becomes breakable. In this article, we presented a new protocol based on the principle of Diffie Hellman [7] and the regenerator of salt RGSCS [2] cryptographically secure and per session. Our principal objective, however, was to protect users even with weak passwords. This leads us to use

these techniques to face the current known attacks by Kerberos V5 such as dictionary attack [33]. Our authentication scheme provides a more reliable model with uncorrelated authentication parameters between different clients in the same realm even if they have identical passwords. This is proved by the behavioral study who presented an encouraging results with unpredictable keys even with the use of a weak password.

References

[1] Y. Asimi, A.Amghar, A. Asimi, and Y. Sadqi, "Strong zero-knowledge authentication based on the session keys (SAAK)," *International Journal of Network Security & Its Applications*, vol. 7, no. 1, p. 51, 2015.

[2] Y. Asimi, A.Amghar, A. Asimi, and Y. Sadqi, "New random generator of a safe cryptographic salt per session," *International Journal of Network Security*, vol. 18, no. 3, pp. 445–453, 2016.

[3] Y. Asimi, A. Amghar, A. Asimi, and Y. Sadqi, "Strong zero-knowledge authentication based on virtual passwords," *International Journal of Network Security*, vol. 18, no. 4, pp. 601–616, 2015.

[4] S. M. Bellovin and M. Merritt, "Encrypted key exchange: Password-based protocols secure against dictionary attacks," in *IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 72–84, Oakland, May 1992.

[5] D. Boneh, *The Decision Diffie-Hellman Problem*, pp. 48–63, Springer, Berlin, Heidelberg, 1998.

[6] D. Cash, E. Kiltz, and V. Shoup. *The Twin Diffie-Hellman Problem and Applications*, pp. 127–145, Springer, Berlin, Heidelberg, Apr. 2008.

[7] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[8] P. Dourish, R. E. Grinter, J. D. De La Flor, and M. Joseph, "Security in the wild: user strategies for managing security as an everyday, practical problem," *Personal and Ubiquitous Computing*, vol. 8, no. 6, pp. 391–401, 2004.

[9] E. El-Emam, M. Koutb, H. Kelash, and O. Faragallah, "An authentication protocol based on kerberos 5.," *International Journal of Network Security*, vol. 12, no. 3, pp. 159–170, 2011.

[10] N. Fazio, R. Gennaro, I. M. Perera, and W. E. Skeith, *Hard-Core Predicates for a Diffie-Hellman Problem over Finite Fields*, pp. 148–165, Springer, Berlin, Heidelberg, 2013.

[11] S. Gaw and E. W. Felten, "Password management strategies for online accounts," in *Proceedings of the second symposium on Usable privacy and security*, pp. 44–55, Pittsburgh, Pennsylvania, USA, July 2006.

[12] J. Kohl and C. Neuman, "The kerberos network authentication service (v5)," Tech. Rep. RFC 1510, Sep. 1993.

[13] J. Y. Kohl, B. C. Neuman, and Y. Theodore, "The evolution of the kerberos authentication service," 1994.

[14] C. C. Lee, C. H. Liu, and M .S.Hwang, "Guessing attacks on strong-password authentication protocol," *International Journal of Network Security*, vol. 15, no. 1, pp. 64–67, 2013.

[15] R. Lidl and H. Niederreiter, "Finite fields: Encyclopedia of mathematics and its applications," *Computers and Mathematics with Applications*, vol. 7, no. 33, p. 136, 1997.

[16] N. Mavrogiannopoulos, A. Pashalidis, and B. Preneel, "Toward a secure kerberos key exchange with smart cards," *International Journal of Information Security*, vol. 13, no. 3, pp. 217–228, 2014.

[17] M.Backes, I. Cervesato, A. D. Jaggard, A. Scedrov, and J. K. Tsay, "Cryptographically sound security proofs for basic and public-key kerberos," *International Journal of Information Security*, vol. 10, no. 2, 2011.

[18] A. Melnikov, "The kerberos v5 (gssapi) simple authentication and security layer (SAAL) mechanism," Nov. 2006.

[19] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC press, 1996.

[20] R. Morris and K. Thompson, "Password security: A case history," *Communications of the ACM*, vol. 22, no. 11, pp. 594–597, 1979.

[21] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks of computers," *Communications of the ACM*, vol. 21, no. 12, pp. 993–999, 1978.

[22] G. Notoatmodjo and C. Thomborson, "Passwords and perceptions," in *Proceedings of the Seventh Australasian Conference on Information Security*, pp. 71–78, Wellington, New Zealand, Jan. 2009.

[23] D. P.Jablon, "Extended password key exchange protocols immune to dictionary attack," in *Sixth IEEE Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp. 248–255, June 1997.

[24] K. Raeburn, "Encryption and checksum specifications for kerberos 5," Feb. 2005.

[25] K. Raeburn, "Network working group c. neuman request for comments: 4120 USC-ISI obsoletes: 1510 t. Yu category: Standards track s. hartman," July 2005.

[26] Y. Sadqi, A. Asimi, and Y. Asimi, "A cryptographic mutual authentication scheme for web applications," *arXiv preprint arXiv:1412.2908*, 2014.

[27] Y. Sadqi, A. Asimi, and Y. Asimi. "Short: A lightweight and secure session management protocol,". in *Networked Systems*, pp. 319–323. Springer, Marrakech, Morocco, May 2014.

[28] B. Schneier, *Applied cryptography: protocols, algorithms, and source code in C*, john wiley and sons, 2007.

[29] J. G. Steiner, B. C. Neuman, and J. I. Schiller, "Kerberos: An authentication service for open network systems," in *USENIX Winter*, pp. 191–202, Dallas, TX, Feb 1988.
[30] Z. Tbatou, A. Asimi, Y. Asimi, and Y. Sadqi, "Kerberos v5: Vulnerabilities and perspectives," in *Third World Conference on Complex Systems (WCCS'15)*, pp. 1–5, Marrakech, Morocco, Nov. 2015.
[31] J. K. Tsay, *Formal Analysis of the Kerberos Authentication Protocol*, PhD thesis, University of Pennsylvania, 2008.
[32] R. Tso, "Security analysis and improvements of a communication-efficient three-party password authenticated key exchange protocol," *The Journal of Supercomputing*, vol. 66, no. 2, pp. 863–874, 2013.
[33] T. D. Wu, "A real-world analysis of kerberos password security.," in *NDSS*, Feb. 1999.
[34] Q. Xie, B. Hu, and T. Wu, "Improvement of a chaotic maps-based three-party password-authenticated key exchange protocol without using server's public key and smart card," *Nonlinear Dynamics*, vol. 79, no. 4, pp. 2345–2358, 2015.

Biography

Tbatou Zakariae received his Master's degree in Computer Science and Distributed Systems in 2013 from Departments of Mathematics and Computer Science, Faculty of Science, University Ibn Zohr, Agadir, Morocco. He is currently Ph.D student in Departments of Mathematics and Computer Sciences, Information Systems and Vision Laboratory, Morocco. His research interests include Authentication Protocols, distributed systems, cloud computing, Computer and Network Security and Cryptography.

ASIMI Ahmed is a full professor at the Faculty of Science, Agadir, Morocco. He received his Ph.D degree in Number theory from Department of Mathematics, Faculty of Science, University Mohammed V, Agdal in 2001, Morocco. He is reviewer at the International Journal of Network Security (IJNS) and at the journal of Computer and Information Science. He is a speaker in national and international conferences on the topics of cryptology and computer security. His main areas of research interests include Number theory, Code theory, Computer Cryptology, Computer and Network Security.

Younes Asimi received his Ph.D. in Strong Zero-Knowledge Authentication Based on virtual passwords per session and the Session Keys in 2015. He is currently pursuing Ph.D in Departments of Mathematics and Computer Sciences, Information Systems and Vision Laboratory, Morocco. His research interests include Authentication Protocols, Computer and Network Security and Cryptography.

Yassine SADQI received his Ph.D in the security of Computer Science and Distributed Systems at the Ibn Zoher University in 2015. Agadir, Morocco. His main ?eld of research interest is computer security, cryptography and authentication in Web applications.

Guezzaz Azidine received his Master's degree in the field of Computer Science and Distributed Systems in 2013 from Departments of Mathematics and Computer Science, Faculty of Science, University Ibn Zohr, Agadir, Morocco. He is currently Ph.D student in Departments of Mathematics and Computer Sciences, Information Systems and Vision Laboratory, Morocco. His main field of research interest is Intrusion Detection and Prevention, Computer and Network Security and Cryptography.