

ATTACKS AND COUNTERMEASURES OF HASH CHAIN BASED PASSWORD AUTHENTICATION SCHEMES



By

Muhammad Shahzad Jan

JUNE 2015

A thesis submitted to the faculty of Information Security Department Military College of Signals, National University of Sciences and Technology, Rawalpindi in partial fulfillment of the requirements for the degree of MS in Information Security

SUPERVISOR CERTIFICATE

IT IS CERTIFIED THAT THE FINAL COPY OF THESIS HAS BEEN
EVALUATED BY ME, FOUND AS PER SPECIFIED FORMAT AND ERROR
FREE.

DR. MEHREEN AFZAL

ABSTRACT

The constant influence of passwords more than all other scheme of identity verification is a key discomfiture to security examiners. As web technologies are gaining more and more fame day by day but the persistent survival and replication of password authentication schemes generate difficulties for end users. After such a long age, still discussions over substitute schemes have not yet produced an ultimate solution. Leslie Lamport suggested the use of hash chains as password verifiers for identity verification over insecure network. Giving importance to the algorithm and structure used in these schemes, W. C. Ku in 2004 proposed a new protocol which has gained so much attraction from researchers till date. This protocol has been exploited and then improved many times. In this research, some attacks on its improved version of protocol have also been demonstrated. Many strong password authentication schemes have been proposed which are based on lamport's method, but none is secure enough. The main aim of this research is to compile a framework for the assessment of hash based password authentication schemes. In this research, a review of strong password authentication schemes has been presented and a suitable new strong password authentication scheme based on lamport's method of hash has been proposed that can fulfill highest possible level of desired criteria according to the framework. An analysis and comparisons of features and security of the proposed scheme with the existing schemes is also included in this research.

DEDICATION

“In the name of Allah, the most Beneficent, the most Merciful”

I dedicate this thesis to my family, friends and honorable teachers.

ACKNOWLEDGEMENTS

Thank God for guiding me to get through it. I wish to express my gratitude to Dr. Mehreen Afzal, my advisor, for her encouragement, insightful guidance and patience throughout my Master degree at Military College of Signals, National University of Science and Technologies. I would like to thank and extend great appreciation to Lt Col. Dr. Babar Aslam, Asst. Prof. Mian Muhammad Waseem Iqbal and Lec. Waleed Bin Shahid for their valuable feedback and professional teaching for preparing this thesis.

Table of Contents

ABSTRACT.....	iii
ACKNOWLEDGEMENTS.....	v
LIST OF FIGURES	xi
LIST OF TABLES.....	xii
KEY TO ACRONYMS	xiii
1 INTRODUCTION	1
1.1 Introduction.....	1
1.2 Incentive.....	1
1.3 Background	2
1.4 Hash Based Password Authentication Protocols.....	3
1.5 Problem Statement	3
1.6 Objectives.....	3
1.7 Sketch of the research	4
1.8 Conclusions	4
2 LITERATURE REVIEW AND BACKGROUND OF STUDY.....	6
2.1 Introduction.....	6
2.2 Existing Studies.....	6
2.2.1 From 1981 to 2000 AD.....	6
2.2.2 From 2000 to 2005 AD.....	7
2.2.3 From 2005 to 2014 AD.....	9

2.3	Identity Authentication over Insecure Network	11
2.4	Something-Known Based Identity Authentication	12
2.4.1	Password Based Authentication.....	12
2.4.2	Challenge Response Based Authentication	12
2.5	Something-Known Based Authentication Methods.....	12
2.5.1	Conventional Passwords	12
2.5.2	Keystroke Dynamics.....	12
2.5.3	Click Patterns	13
2.5.4	Graphical Passwords.....	13
2.5.5	Authentication Panel.....	14
2.5.6	Reformation Based.....	15
2.5.7	Moving Balls Based.....	15
2.5.8	Expression Based	15
2.6	Types of Password Based Authentication.....	17
2.6.1	RSA-Based Password Authentication Schemes	17
2.6.2	ElGamal-Based Password Authentication Schemes.....	17
2.6.3	Hash-Based Password Authentication Schemes.....	17
2.7	Conclusions	19
3	PROPOSED FRAMEWORK	20
3.1	Introduction	20

3.1	Goals.....	20
3.2	Benefits.....	21
3.2.1	Usability Benefits.....	21
3.2.2	Deployability Benefits	22
3.3	Security Requirements	23
3.4	Flow Diagram.....	26
3.5	Conclusions	28
4	ANALYSIS AND COMPARISONS OF RENOWNED PROTOCOLS	29
4.1	Introduction	29
4.2	Optimal Strong Authentication Protocol (OSPA).....	29
4.2.1	Registration Phase.....	31
4.2.2	Authentication Phase	31
4.3	Enhanced- OSPA (E-OSPA).....	33
4.3.1	Registration Phase.....	33
4.3.2	Authentication Phase	34
4.4	W.C. Ku's Password Authentication Scheme	35
4.4.1	Registration Phase.....	36
4.4.2	Login Phase.....	37
4.4.3	Kim-Koc's Attacks on W.C. Ku's Scheme	39
4.4.4	Security Weaknesses of W. C. Ku's Scheme by M. Kumar.....	40

4.4.5	Yang-Shen's Comments on W. C. Ku's Scheme	43
4.4.6	Proposed Attacks on W.C. Ku's Scheme.....	43
4.4.7	Analysis of W. C. Ku's Scheme	45
4.5	CompChall Authentication Protocol	47
4.5.1	Review of CompChall Protocol.....	48
4.5.2	Proposed Attacks on CompChall Protocol	51
4.6	Conclusions	53
5	PROPOSED SCHEME.....	54
5.1	Introduction	54
5.2	Hash Chain based Strong Password Authentication Scheme	54
5.3	Registration Phase	56
5.4	Login Phase	58
5.5	Password Change Phase	61
5.6	Conclusions	63
6	FRAMEWORK SCRUTINY OF PROPOSED SCHEME.....	64
6.1	Introduction	64
6.2	Analysis and Comparison of Proposed Scheme.....	64
6.3	Conclusions	69
7	CONCLUSIONS AND FUTURE WORK	70
7.1	Introduction	70

7.2	Conclusions	70
7.3	Future Work	71
	BIBLIOGRAPHY	73

LIST OF FIGURES

Figure 2.1: Types of Identity Authentication.....	11
Figure 2.2: Keystroke Dynamics example.....	13
Figure 2.3: Click Patterns example.....	13
Figure 2.4: Graphical Passwords example.....	14
Figure 2.5: Authentication Panel example.....	15
Figure 2.6: Example of Expression based Authentication.....	16
Figure 2.7: Structure of Lamport's Hash Chains.....	19
Figure 2.8: Generation of OTP	19
Figure 3.1: Flow Chart for Testing Ideal Password Authentication Scheme	27
Figure 4.1: Registration Phase of OSPA Protocol.....	31
Figure 4.2: Authentication Phase of OSPA Protocol.....	32
Figure 4.3: Registration Phase of E-OSPA Protocol	34
Figure 4.4: Authentication Phase of E-OSPA Protocol.....	35
Figure 4.5: Registration Phase of W.C.Ku's Scheme	37
Figure 4.6: Login Phase of W. C. Ku's Scheme	38
Figure 4.7: CompChall Authentication Protocol 1	49
Figure 4.8: CompChall Authentication Protocol 2	51
Figure 4.9: Attacks on CompChall Authentication Protocol.....	53
Figure 5.1: Registration Phase of Proposed Scheme	57
Figure 5.2: Login Phase of Proposed Scheme	60
Figure 5.3: Password Change Phase of Proposed Scheme	62

LIST OF TABLES

Table 2.1: Comparison between Authentication Methods.....	16
Table 4.1: 'Goals' Analysis of W. C. Ku's Scheme.....	45
Table 4.2: 'Benefits' Analysis of W. C. Ku's Scheme.....	46
Table 4.3: 'Security Requirements' Analysis of W. C. Ku's Scheme.....	46
Table 6.1: Analysis and Comparisons	69

KEY TO ACRONYMS

DOS	Denial-of-Service
SV	Stolen-Verifier
MiTM	Man-in-the-Middle
OSPA	Optimal Strong Password Authentication
CompChall	Computation Challenge
SHA	Secure Hash Algorithm
SAS	Simple and Secure Password Authentication Protocol
PERM	Privacy Enhanced Information Reading and Writing Management Protocol
DoS	Denial of Service
SPAS	Strong Password Authentication Scheme
CAPCHA	Completely Automated Public Turing Test to Tell Computers and Humans Apart
OTP	One-Time-Password
E-OSPA	Enhanced Optimal Strong Password Authentication

INTRODUCTION

1.1 Introduction

Authentication schemes based on strong passwords have been commonly installed to confirm the authenticity of remote clients. Authentication based on strong passwords is very much simple and the most handy authentication system for remote and insecure networks. Hash based strong password verification schemes are based on one-way functions having a challenge-response technique and are preferred in most of the scenarios because of its better usability, scalability and reliability qualities with low communication and computational cost.

In this research a framework have been compiled considering goals, benefits and security requirements needs to carry out for an ideal hash based password authentication scheme. A combination of Man-in-the-Middle (MiTM) attack, Stolen-Verifier (SV) attack and DoS attack on ad previously proposed scheme [14] have been demonstrated and proposed an enhanced scheme over W. C. Ku's scheme. Proposed scheme is based on hash chain mechanism using smart card. The scheme has the capability to resist most of the attacks, achieve the major security and functionality goals, and provide benefits of usability and deployability according to the proposed framework.

1.2 Incentive

Web technologies are gaining more and more fame day by day but the persistent survival and replication of password authentication schemes generate

difficulties for end users. After such a long age, still discussions over substitute schemes have not yet produced an ultimate solution.

New generations have been grown up with today's elevated digital technologies. Every citizen including government personals, employees, researchers, doctors, engineers and students have been extensively involved in technologies specially in Web based like social networking sites, office automation systems, electronic offices, e-commerce, banking, hospital and pharmacy management systems, etc, either at work or at home, and every other one is just at a distance of typing few words. Ignorant of the core technologies, and regardless of their momentum of sharing, they are in their own irregular way of sharing sensitive data via insecure paths without having real and transparent authentication of the system and its users.

1.3 Background

The most preferred method for using hash based authentication is Lamport's hash chain based method [1] introduced in 1981. Based on Lamport's method W. C. Ku proposed a more secure scheme without using smart card. Ku's scheme has been considered to be better to numerous renowned protocols like CINON [2], S/KEY [3], PERM [5], SAS [6], OSPA [7], E-OSPA [10] and ROSI [12]. However, Kim and Koc [17] demonstrated SV attack, DOS attack, replay attack and impersonation attack on Ku's scheme. Afterwards, M. Kumar [34] found some security weaknesses in the protocol and demonstrated insider attack, parallel session attack, guessing attack, MiTM attack, SV attack, impersonation attack and DOS attack on Ku's scheme. Yang and Shen [35] suggested some modifications for the improvement of Ku's scheme to resist SV attack. More recently, Zhuang, Chang and Wang [40] presented MiTM

attack on Ku's scheme and proposed a geometric hashing based scheme without using smart card.

1.4 Hash Based Password Authentication Protocols

Leslie Lamport suggested the use of hash chains for identity verification over insecure network. In 2001, Optimal Strong Authentication Protocol (OSPA) was proposed and remained under consideration for the authentication mechanism over insecure network so far. OSPA has been demonstrated vulnerable by many authors and has been improved time by time. In 2003, an Enhanced OSPA (E-OSPA) protocol was proposed but was again proved to be vulnerable for different types of attacks. Giving importance to the algorithm and structure used in these schemes, W. C. Ku in 2004, proposed a new protocol which has gained so much attraction from researchers till date. This protocol has been exploited and then improved many times. In our research we have also demonstrated some attacks on its improved version of protocol.

1.5 Problem Statement

Password based authentication schemes using hash chains (One Time Password) are quite desirable and preferred the most for many scenarios. OSPA protocol, E-OSPA protocol and W. C. Ku's scheme has been center of attention for so many years, but it is also found vulnerable to different types of attacks that are discussed in this research. Due to their usability, however, improvement of hash based password authentication schemes is still an OPEN research area.

1.6 Objectives

A framework on hash chain based password authentication schemes has been discussed in different researches. The objective of this research is to:

1. To analyze and compose a framework to highlight the merits and demerits of different secure hash chain based password authentication schemes.
2. To design a suitable protocol that can fulfill highest possible level of desired criteria of given framework.
3. To confirm the security features of the projected scheme in comparison with the existing schemes.

1.7 Sketch of the research

This thesis has been structured as follows: Chapter 2 represents a complete flow of existing schemes and literature review. Chapter 3 defines desired criteria in form of a proposed framework with designed flow diagram for assessment of authentication schemes. In Chapter 4, OSPA, E-OSPA and W. C. Ku's scheme have been reviewed and proposed attacks have been demonstrated with complete analysis and comparisons. While in Chapter 5, the proposed scheme has been presented and the framework scrutiny of proposed scheme has been shown in Chapter 6. Conclusions and future work has been finally drawn at the end.

1.8 Conclusions

Password based authentication schemes are used in many platforms because of its simplicity, usability and deployability. Hash chains are used to generate one time passwords for the authentication of legitimate users for every session over insecure network. Many schemes based on strong password have been proposed but neither of them has successfully achieved security requirements. In this thesis, a complete framework has been proposed for the analysis of hash chain based password authentication schemes. Compared to renowned authentication schemes, a better and

suitable scheme has been proposed that accomplishes highest possible level of desired criteria of proposed framework.

LITERATURE REVIEW AND BACKGROUND OF STUDY

2.1 Introduction

In this chapter, complete flow of different schemes and algorithms are discussed in detail. Approaches used for identity authentication over insecure network, are also discussed. According to the scope of this research, the flow of schemes since 1981 to 2014 AD is being discussed in a form of hierarchy. Methods used for something known based identity authentication and hash based password authentication methods are also discussed in this chapter.

2.2 Existing Studies

Several schemes have been proposed since Lamport's hash chain method being introduced in 1981. The flow of schemes is divided into three different periods according to the difference in approaches and algorithms used in series of schemes.

2.2.1 From 1981 to 2000 AD

In 1981, Lamport [1] discussed general problems of authentication over insecure network and first time introduced a hash chain method for password based authentication. A. Shimizu [2] encountered some drawbacks in Lamport's scheme and proposed a new scheme called CINON in 1990. In CINON, first time two variable random numbers were used for each session and One-Time-Password (OTP) characteristic was gained in this protocol. Haller [3] in 1995 discussed some security flaws of Lamport's scheme and demonstrated a replay attack on the protocol.

Furthermore, he proposed an S/Key OTP system. Chen-Mitchell [4] again demonstrated replay attack on S/Key OTP system in 1996. A. Shimizu [5] again proposed a new protocol based on lamport's method named PERM. This time A. Shimizu used one variable random number unlike CINON. It was more secure and computationally efficient than CINON. SANDIRIGAMA-SHIMIZU [6] discussed security flaws of S/Key OTP System, CINON & PERM and demonstrated a MiTM attack on PERM protocol. Furthermore, they proposed a new protocol called SAS (Simple and Secure) password authentication protocol based on Lamport's method having reduced storage, processing and transmission overhead in 2000.

2.2.2 From 2000 to 2005 AD

In 2001, LIN-SUN-HWANG [7] demonstrated replay attack and DOS attack on SAS protocol and proposed a new protocol named OSPA protocol and claimed that their scheme can defend against against replay attack, DOS attack and SV attack. But CHEN-KU [8] in 2002 and TSUJI-SHIMIZU [9] in 2003 have demonstrated SV attack and impersonation attack on OSPA protocol respectively.

In 2003, Lin et al. [10] proposed new protocol named Enhanced-OSPA (E-OSPA) protocol. Server's secret key and smart card have been used for the first time in this protocol. They claimed that their scheme can defend against against guessing attack, replay attack, impersonation attack and SV attack. However, Ku et al. [11] demonstrated replay attack and DOS attack on E-OSPA.

At the same time, CHEN-JAN [12] discussed security weaknesses of SAS and OSPA protocols and proposed ROSI protocol and claimed to be safe against MiTM attack, SV attack and guessing attacks. But within one year KU-TSAI-TSAUR [13] proved security weaknesses of ROSI protocol.

In 2004, Ku [14] came up with a new protocol on the basis of enhanced OSPA protocol but without using smart card. Instead on smart card, he has used timestamp for his protocol. The protocol was claimed to be safe against replay attack, DOS attack, password-file compromise attack, forgery attack and predictable 'n' attack. Easy reparability was ensured in this protocol.

CHEN-LEE-HORNG [15] discussed security weakness and demonstrated DOS attack on enhanced OSPA. They proposed Secure-SAS like scheme which was considered secure against user impersonation attack, server spoofing attack, replay attack, SV attack and DOS attack. Further, CHANG-CHANG [16] demonstrated server spoofing attack and DOS attack on SAS, OSPA and E-OSPA protocols. They have also proposed new and improved scheme.

In 2005, KIM-KOC [17] demonstrated SV Attack, DOS Attack, replay attack and impersonation attack on W. C. Ku's scheme and CHIEN-WANG-YANG [18] raised state synchronization property problem in ROSI protocol. They demonstrated DOS attack and proposed improved version of ROSI protocol.

CHANG-CHANG [19] proposed improved version of SAS and OSPA protocols which provides mutual authentication property which were not addressed on previous versions. At the same time, YOON-RYN-YOO [20] discussed about comparisons of security problems of SAS, OSPA & E-OSPA protocols and proposed a newer version of E-OSPA protocol.

WU-HWANG-LIU [21] proposed Securer OSPA Protocol and YOON-YOO [22] proved that Secure SAS-like protocol is not easily reparable. They defined Security Theorems for authentication schemes and demonstrated DOS Attack and insider attack on Secure SAS-like protocol and proposed RSK Authentication

Scheme. While, V. Goyal et al. [23] proposed a new scheme named CompChall authentication protocol which resist against dictionary attacks.

2.2.3 From 2005 to 2014 AD

In 2006, LIN-TSAI-HWANG [24] proposed SPAS protocol while encountering security flaws of E-OSPA and attacks demonstrated in previous versions. CHANG-TSAI [25] performed SV attack on CHANG-CHANG's protocol and proposed Enhanced CHANG-CHANG's protocol. MANGIPUDI-KATTI [26] claimed that one common feature of the hash-based authentication protocols is that the client's identity is transmitted in plain during the authentication process, which allows an attacker to monitor the user activities. They proposed SPAPA protocol which provides user anonymity and provide security against guessing attack, SV attack, replay attack, stolen smart card attack and DOS attack. TSAI-LEE-HWANG [27] defined goals to be achieved by every password authentication scheme and attacks to withstand. They compared different protocols against goals and security requirements.

In 2007, MITCHELL-SIAW [28] claimed that as SPAPA protocol was intended to defend users against eavesdropping attacks by utilizing impermanent identities as an alternative of proper identities, but it is still exposed to several attacks. They demonstrated MiTM attack, replay attack and SV attack. They also raised synchronization issue of SPAPA protocol. TIAN-ZHU-WONG [29] proposed two new schemes and claimed that their scheme is performance wise more competent than E-OSPA. SANDIRIGAMA-WERAGAMA [30] proposed SAS-3 protocol and comparisons between SAS-3 and previously discussed protocols.

In 2009, SOOD-SARJE-SINGH [31] analyzed previously discussed protocols and their loopholes. They drew a comparison table that provides complete picture of security features of different protocols. LEE-WOU [32] discussed security flaws of E-OSPA, Secure SAS-like and SPAS protocol. They also defined some security requirements for authentication schemes and performed guessing attack and impersonation attack on Secure SAS-like and SPAS protocols. They proposed Enhanced Secure-SAS Like protocol. At the same time, LIANG-ZHIGANG [33] proposed Low Cost OTP scheme.

In 2010, MANOJ KUMAR [34] discussed that session key generation and password change phases and mutual authentication is not provided in Ku's scheme. He demonstrated few attacks on Ku's scheme. Yang and Shen [35] performed MiTM attack on Ku's scheme and suggested some modifications for the improvement to resist SV attack. HUIPING [36] proposed SPAS extension of E-OSPA which withstands replay and DOS attacks of E-OSPA.

In 2012, J. Bonneau et al. [37] presented a technical report on different OTP authentication protocols. They discussed different security flaws of different password based authentication schemes introduced so far. They provided Usability, Deployability and Security requirements of an ideal password authentication protocol to fulfill.

In 2013, LEE-LIU-HWANG [38] performed guessing attack on OSPA protocol and ELKAMCHOUCHE-ELDEFRAWY [39] demonstrated theft attack and impersonation attack on Securer OSPA protocol. In 2014, X. Zhuang et al. [40] performed MiTM attack on W. C. Ku's attack and proposed Geometric Hashing based scheme without using smart card.

2.3 Identity Authentication over Insecure Network

Authentication guarantees that assets are not obtained deceitfully by an illegal entity. Identity Authentication over insecure network can be broken down into three different factors, casually known as "*something you have*", "*something you know*", and "*something you are*". Token-based authentication or Smart Card based authentication (something you have) is normally a physical device such as a USB dongle, debit card or security card. Knowledge-based authentication (something you know) could be a password, or it could involve questions about the user's life, basically includes challenge response method. Biometrics or Characteristics based authentication (something you are) can be further divided into physiological characteristics, like a fingerprint or retinal scan, and behavioral characteristics, such as a handwritten signature or a voice sample. The hierarchy of identity authentication over insecure network has been shown in Figure 2.1.

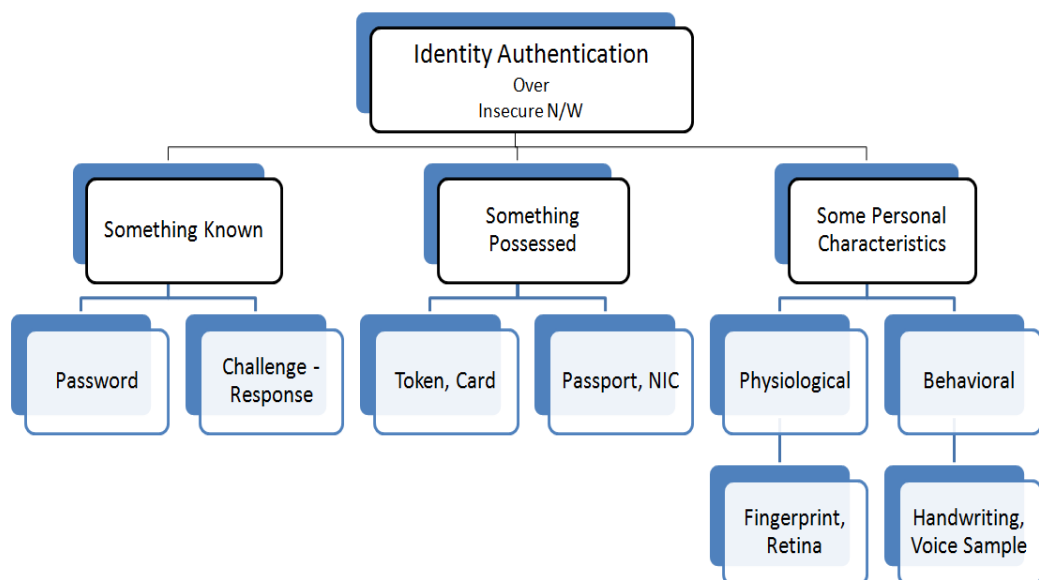


Figure 2.1: Types of Identity Authentication

2.4 Something-Known Based Identity Authentication

2.4.1 Password Based Authentication

It is one of the most critical practical functions that we all use every day, yet it hasn't advanced much. According to Moore's Law, which continually provides faster processors to crack password databases in less time, today's password based authentication schemes insecurely provide open hand for brute force or dictionary attacks.

2.4.2 Challenge Response Based Authentication

This method is used for authenticating over an insecure network without passing any secret out to eavesdroppers that may allow attackers to authenticate as you.

2.5 Something-Known Based Authentication Methods

2.5.1 Conventional Passwords

In this method, end user passes his/her credentials and server confirms user's identity from the database using user provided username and password, and then allows access to the resources.

2.5.2 Keystroke Dynamics

This scheme deals with 'HOW' the client has passed his/her credentials instead of confirming with 'WHAT' the user has passed through. It records the key press and key timings.

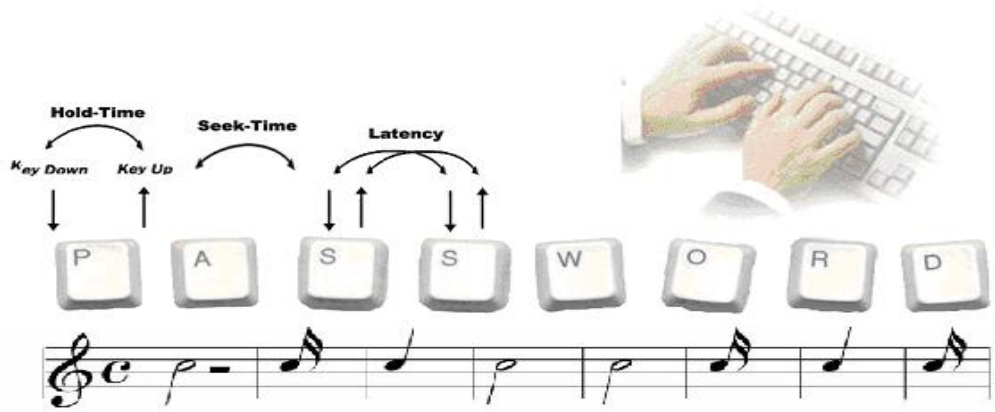


Figure 2.2: Keystroke Dynamics example

2.5.3 Click Patterns

In this method, the user passes a specific pattern from the colored click pad or a grid. Dry fingers can reveal password in case of touch screen as shown in figure 2.3.

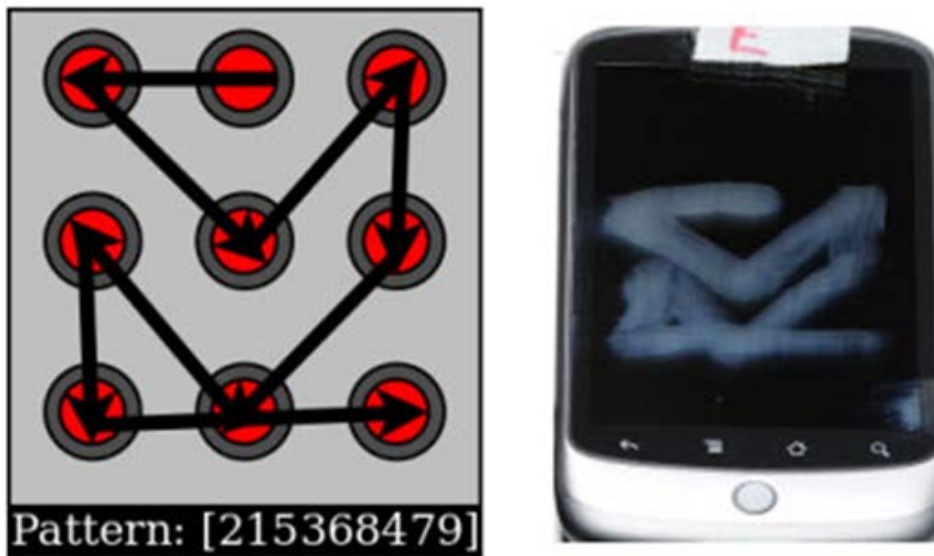


Figure 2.3: Click Patterns example


2.5.4 Graphical Passwords

In this method, after passing username, server asks user to draw a selected graphical sketch using input device from list of objects that would be shown by

server. Server performs some processing and matches the user drawn sketch with database.

Username:

Password:

				
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
				
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
				
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
				
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
				
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

First Name:

Last Name:

Gender: Male Female

Birthday:

country:

Figure 2.4: Graphical Passwords example

2.5.5 Authentication Panel

In this method, user doesn't need to press the button for passing secret to the server; instead, user has to select the location of the secret from provided panel. It provides security against several attacks.

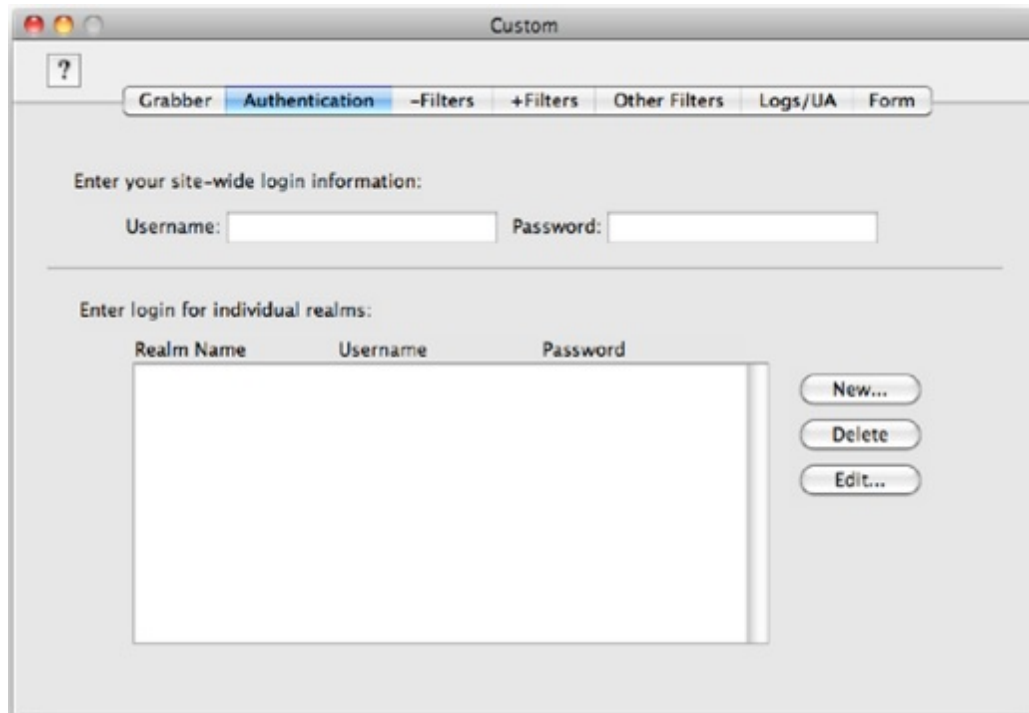


Figure 2.5: Authentication Panel example

2.5.6 Reformation Based

In this method the user provided secret is converted to new shape before keeping it in database and whenever the secret has to be study then it must be necessary to reconvert for verification.

2.5.7 Moving Balls Based

In such method, server provides various balls moving in different directions horizontally or vertically in columns, client has to memorize directions of moving balls and their particular columns.

2.5.8 Expression Based

In this method, user has to memorize and generate secret based on expressions and password that are provided by server.

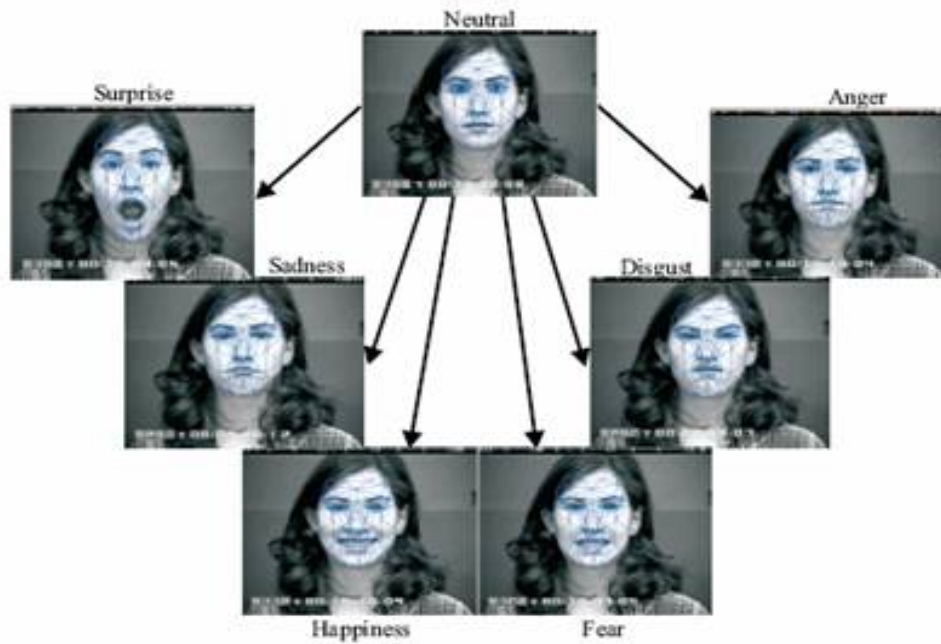


Figure 2.6: Example of Expression based Authentication

Different methods can be combined together to create a single and more secure method. Like, Conventional + Keystrokes Dynamics, Conventional + Click Patterns and Conventional + Expression based etc. The comparisons between different authentication methods have been shown in Table 2.1.

Table 2.1: Comparison between Authentication Methods

Method	Additional H/W Requirements	Cost	Mental Attitude	Protection Level	Processing Time
CONVENTIONAL	NO	NORMAL		LOW	FAST
KEYSTROKE DYNAMICS	NO	NORMAL	YES	MEDIUM	MEDIUM
CLICK PATTERNS	NO	NORMAL	YES	MEDIUM	MEDIUM
GRAPHICAL PASSWORDS	YES	HIGH	YES	MEDIUM	SLOW
AUTHENTICATION PANEL	NO	NORMAL	YES	HIGH	MEDIUM
REFORMATION BASED	NO	NORMAL	NO	MEDIUM	FAST
MOVING BALLS BASED	NO	NORMAL	YES	HIGH	MEDIUM
EXPRESSION BASED	NO	NORMAL	YES	HIGH	FAST

2.6 Types of Password Based Authentication

Literature has shown that three types of password authentication schemes are most commonly in use and these are

2.6.1 RSA-Based Password Authentication Schemes

These types of schemes are the first practical public-key cryptosystem based schemes and are broadly used for identity verification and communication. In such schemes both the encryption and decryption keys are different from each other and decryption key is always kept secret. RSA is an algorithm for public-key cryptography that is based on the problem exists in integer factorization.

2.6.2 ElGamal-Based Password Authentication Schemes

These types of schemes are also based on public-key cryptosystem. It consists of both encryption and signature algorithms. ElGamal encryption consists of three components: the key generator, the encryption algorithm, and the decryption algorithm. ElGamal is an encryption scheme that, like RSA, depends on computational assumptions to guarantee security. Unlike the RSA assumption, however, ElGamal depends on type of assumption called the Discrete-Logarithm assumption.

2.6.3 Hash-Based Password Authentication Schemes

These types of schemes are based on one way functions. Hash-based Password Authentication schemes are preferred in most of the scenarios because of its better usability, scalability and reliability with low communication and computational cost. Moreover, unlike other types of schemes, it does not require any trusted third

part and Public Key Infrastructure. User can use dynamic identities and the major problem of replay attacks can be completely eliminated via hash chains.

2.6.3.1 Lamport's Hash Chain Scheme

The most preferred method for using hash based authentication is Leslie lamport's Hash Chain based method [1] introduced in 1981. He provided Hash Chain based solution against below three ways by which an attacker could study the client's secret for masquerading.

1. Studying the server's secret verification file.
2. Eavesdropping on the line.
3. Choosing an easily guessed password.

Solutions: Lamport proposed solutions for the above mentioned problems.

For 1: Use a hashing to convert the secret into different code.

For 2 & 3: Client uses a series of secrets $X_1, X_2, \dots, X_{1000}$ where X_i is the secret by which the client verifies him/herself for the i^{th} instance. The server must use the one-way function to get and use $Y_i = F(X_i)$ and the Y_i must be different to thwart an attacker from replaying a previous secret.

2.6.3.2 Algorithm of Lamport's Scheme

Password X for i^{th} time would be $X_1 = F^{n-i}(X)$ then 'n' no of total passwords would be :

$$F^{n-1}(X), F^{n-2}(X), \dots, F(F(F(X))), F(F(X)), F(X), X.$$

Server Authentication key Y for i^{th} time would be: $Y^{i+1} = F^{(n-i)+1}(X)$

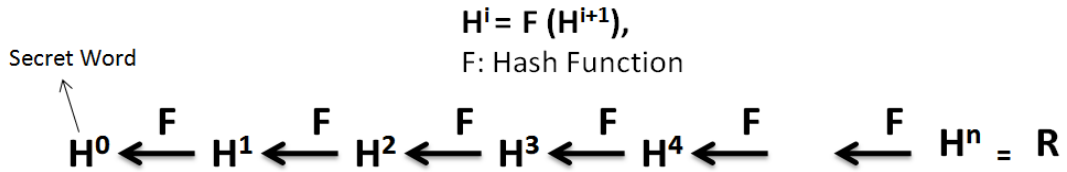


Figure 2.7: Structure of Lamport's Hash Chains

2.6.3.3 Properties of One Way Hash Chains

With given value of H^i , anybody can compute H^j , where $j < i$. It would be computationally infeasible to compute H^I , where $I > i$, if H^x is unknown. Any H^x disclosed later can be authenticated by verifying if $H^{x-i}(H^i) = H^x$. Disclosing of H^{i+1} or a later value authenticates the owner of the hash chain. The properties are illustrated in Figure 2.8.

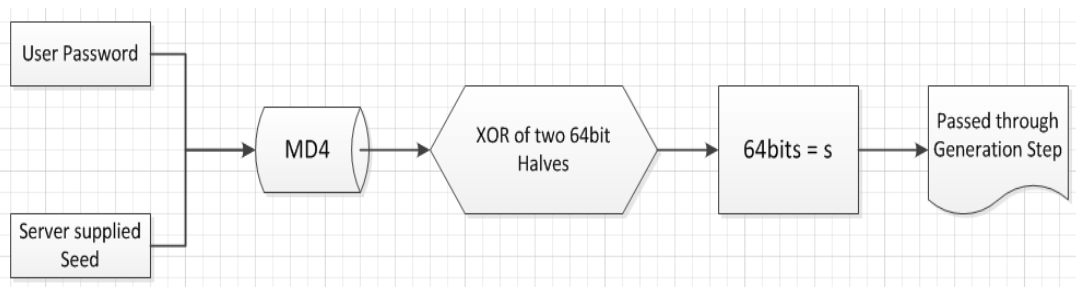


Figure 2.8: Generation of OTP

2.7 Conclusions

Lamport introduced hash chain based method for password authentication in 1981. But since then researchers are not able to propose an ultimate solution. W. C. Ku's scheme is the center of attention since 2004. There are several something known based identity authentication methods being introduced, but password based methods are used mostly in different infrastructures. Furthermore, Lamport's hash chains are used for the security of password based identity authentication.

PROPOSED FRAMEWORK

3.1 Introduction

In this chapter a basic framework for the analysis of hash chain based password authentication schemes has been assembled from different surveys and researches, and presented in such a way that any password based authentication scheme could be examined with reference of our given framework, which has been organized and designed exclusively for testing the strength of password-based authentication schemes. Framework has been divided into three major aspects of authentication schemes.

The framework encompasses three categories: goals, benefits and security requirements. A flow diagram is being designed which should be used for the assessment of hash based password authentication schemes. The framework has been developed to a set that emphasizes significant assessment dimensions, with an eye to prevent overlap between each other. The framework is being composed after analyzing various appraisals and researches [09] [14] [15] [16] [18] [22] [26] [27] [28] [37]. Factors of the framework are referred to with a mnemonic title and a unique notation.

3.1 Goals

An ideal hash based strong password authentication scheme should achieve following goals.

- G1.** User's credentials (passwords and verifiers) should not be stored in plain.

- G2.** Remote client or user can have an option for changing passwords without restraints.
- G3.** The insiders or system administrators cannot reveal user's credentials.
- G4.** Verifiers are not communicated in plain.
- G5.** Users are not bound for choosing password of an inappropriate length.
- G6.** The scheme is robust and practical.
- G7.** Use of wrong credentials for unauthorized login can be detected rapidly.
- G8.** Key for next session is established during authentication phase.
- G9.** The user ID is dynamic for every session to evade outflow of fractional information.
- G10.** The scheme remains secure if somehow server's secret key is compromised.
- G11.** The scheme has a lesser amount of storage and processing requirements.
- G12.** The scheme has a reduced operating cost for transmission over the network.

3.2 Benefits

An ideal hash based strong password authentication scheme should provide following usability and deployability benefits.

3.2.1 Usability Benefits

There are eight usability benefits which are described as under.

UB1 Memory Wise Effortless

Users are not restricted to memorize passwords or secrets more than user's capability.

UB2 User's Resources Wise Affordable

It should not raise burden for maximum number of accounts of user.

UB3 Nothing To Carry

Users do not require holding an extra electronic device, token, piece of paper etc like smart card to use the system.

UB4 Physically Effortless

The verification procedure should not have need of physical user attempt beyond, say, pressing a button.

UB5 Easy To Learn

Users can figure it out, be trained and remember the use of scheme without too much trouble while don't know the scheme.

UB6 Efficient To Use

It should have required practical time for user to spend for login.

UB7 Infrequent Errors

Infrequent errors should be stopped that discourage genuine users by rejecting actual user.

UB8 Easy Recovery From Loss

User should have an ability to recover easily in case of lost or forgetting password.

3.2.2 Deployability Benefits

There are four deployability benefits which are described below.

DB1 Accessible

Users with disabilities or or any other substantial body circumstances should not be prohibited.

DB2 Negligible Cost Per User

The total price per user of the scheme, adding up the expenses at both the applicant's end (any devices required) and the verifier's end (any share of the tools and software required), is minor.

DB3 Server Compatible

The scheme should be compatible with text based passwords at server's end.

DB4 Client Compatible

Users should not have to change or modify their application like browsers, to carry on with the scheme.

3.3 Security Requirements

An ideal hash based strong password authentication scheme should accomplish following security requirements.

SR1 Resilient To Physical Observation.

An attacker should not be able to observe communication between client and server on the flow or during its execution.

SR2 Resilient To User Impersonation Attack.

It should not be feasible for a trained adversary to masquerade as an explicit client by taking advantage of personal information. An attacker shouldn't be able to intercept the communications and modify to masquerade as an authorized user to authenticate.

SR3 Resilient To Server Spoofing Attack.

A person or program should not be able to successfully masquerade as a server to the user by falsifying data and thereby to gain an illegal advantage.

SR4 Resilient To Denial-Of-Service Attack.

Attacker shouldn't be able to keep logging with false verification credentials either by central or distributed ways.

SR5 Resilient To Man-In-The-Middle Attack.

Scheme should have the possessions to stop a malicious actor to have the ability to both observe and modify or inject messages into a communication channel. The attacker shouldn't be able to insert him/herself into the conversation between server and client to imitate any party.

SR6 Resilient To Parallel Session Attack.

An attacker shouldn't be able to use authentication tokens or a valid login message to create session with server in parallel to impersonate as an authorized user.

SR7 Resilient To Replay Attack.

System shouldn't accept a previously communicated message. Attacker should not be able to communicate by replaying previous messages with both server and client.

SR8 Resilient To Guessing Attack.

Passwords should have higher entropy. Attacker shouldn't be able to use guessed password correctly by intercepting authentication messages either by online or offline.

SR9 Resilient To Stolen-Verifier Attack.

If an attacker was somehow able to steal the verifier, should not be able to masquerade as legitimate user to the server.

SR10 Resilient To Insider Attack.

A malicious actor with an authorized system access should not be able to perpetrate or exploit scheme on a network or computer system.

SR11 Resilient To Password-File Compromise Attack

Password-file contains verifiers and credentials thereby to facilitate server for the authentication of a legitimate user. The password-file must be kept secure and out of range from illegitimate access.

SR12 Resilient To Predictable 'N' Attack.

Most of the hash chain based authentication schemes used a unique Nonce often represented by 'N' that increments after every successful login. This Nonce should be kept unpredictable because it leaks sometimes partial and sometimes complete information regarding session keys. This type of attack often became handy for denial-of-service attacks.

SR13 Resilient To Smart-Card Loss Attack.

With stolen or lost smart card, unauthorized users shouldn't be able to easily change new password or exploit system using guessing attacks or impersonation attack.

SR14 User Anonymity.

A common aspect of the hash-based authentication schemes is the transmission of user's identity in plain during authentication process, which allows an adversary to observe user proceedings. Scheme should have ability where an adversary could not identify the user who is trying to authenticate.

SR15 Mutual Authentication.

A client process must confirm its identity to a server, and the server must confirm its identity to the client, before any conversation is being started between both parties. Both the user and server should authenticate each other and this way both parties would be able to recognize each other. This process should occur before continuing with the session.

SR16 Forward Secrecy.

It must be ensured by the system that the previously generated passwords are protected even if the server's secret key has been public by any means.

SR17 No Trusted Third Party.

Scheme must not depend on a trusted third party other than the applicant and verifier.

SR18 State Synchronization.

The state synchronization feature is commonly used for data that needs to be synchronized between client and server for authentication process. This type of property often makes scheme exposed to the DoS attack. The scheme should withstand against state synchronization issues by using strong algorithms or tamper-proof facilities. Every state should be synchronized and confirmed before allowing network traffic because it's an important part of authentication process.

3.4 Flow Diagram

Flow diagram has been designed for the evaluation of hash based password authentication schemes as shown in Figure 3.1. It has been kept with an aspect of proposed framework. Every scheme should be firstly passed from functionality testing and further go by proposed framework. Anywhere between factors of the framework, if a scheme fails to achieve any goal, does not provide any benefit or have not fulfilled any security requirement, a scheme must be refined before passing it for further evaluation process. After executing assessment of the scheme, if a scheme exceed till the end of framework than there should be performed a performance analysis also. In a performance analysis process, scheme must be look upon with preciously available schemes. If a scheme surpasses the predefined threshold then it can be declared an ideal authentication scheme for implementation.

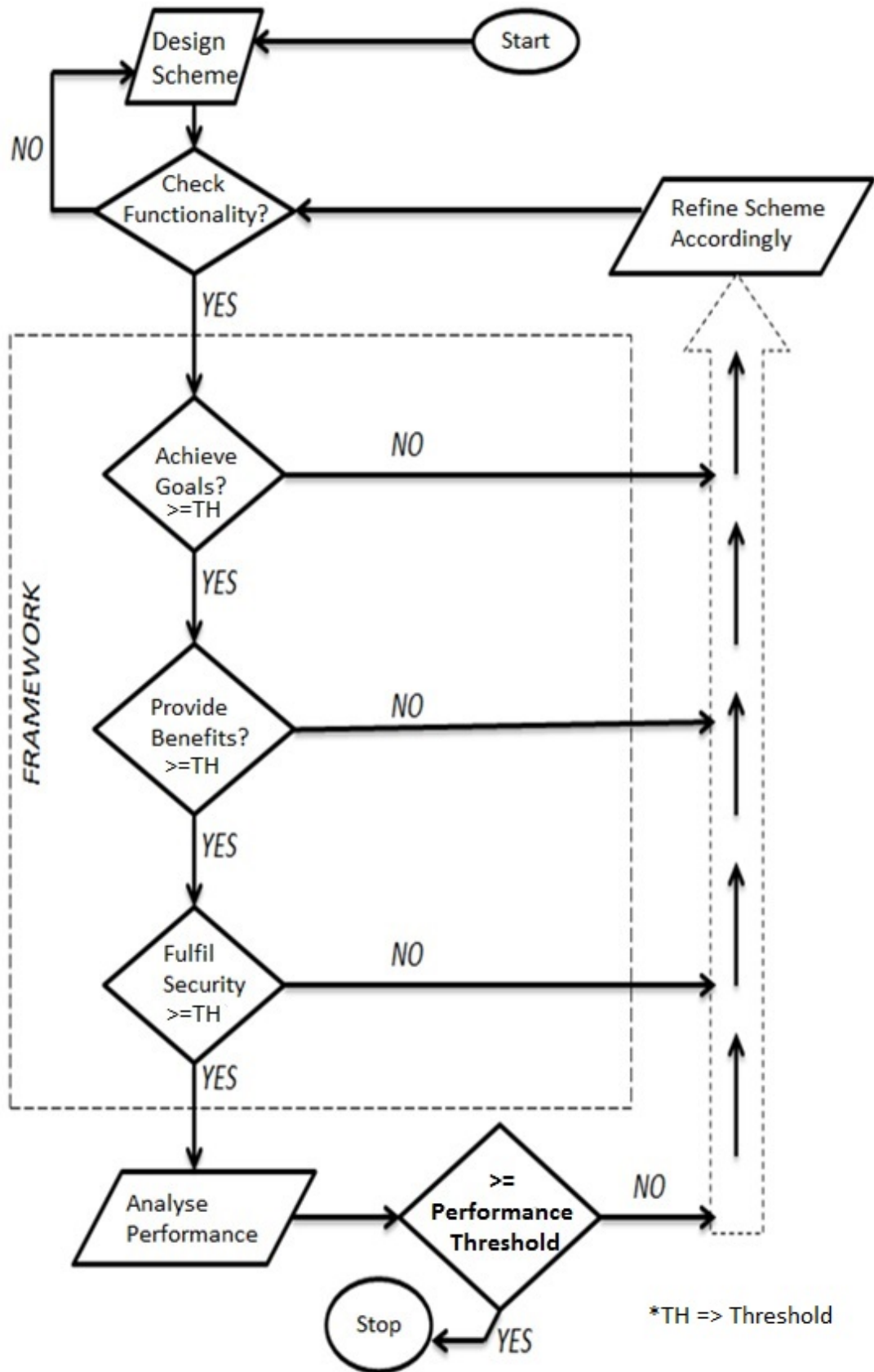


Figure 3.1: Flow Chart for Testing Ideal Password Authentication Scheme

3.5 Conclusions

The proposed framework is divided into three main categories for the analysis of an ideal hash based password authentication scheme. Goals and Benefits consist of twelve points each. While eighteen requirements of security, are arranged in such an order that none of them overlap with any other factor of this category. A complete flow diagram is being presented for the analysis of scheme. The framework is being introduced after considering various reviews.

ANALYSIS AND COMPARISONS OF RENOWNED PROTOCOLS

4.1 Introduction

In this chapter most known hash based authentication scheme have been reviewed and evaluated aligned with the desired framework.

In 1998, A. Shimizu [5] proposed authentication scheme called PERM for e-mail forwarding. The PERM protocol inherited the one-time password method from previous works [1], [2], [3] and [4], in which the high hash overhead was reduced and the password resetting problem was solved. The PERM protocol solved the random number memorization problem caused by the CINON [2] method.

In 2000, SAS protocol was proposed [6], in which the authors pointed out that a kind of MiTM attack can succeed in both CINON and PERM. They claimed that SAS eliminated these types of attacks and has lowered storage and processing requirements and reduced transmission overhead.

4.2 Optimal Strong Authentication Protocol (OSPA)

OSPA was proposed by Lin et al in 2001 as a replacement for SAS. They claimed that their protocol is secure against SV attack, replay attack and DOS attacks which were demonstrated on SAS before. Their scheme used minimum computation, storage and transmission overhead.

Authentication is crucial to the security and accounting of many service systems. Many authentication mechanisms have been developed using biometric techniques but these are not cost effective and suitable for Internet, Mobile Applications and Web Applications. Easy to remember and weak passwords cause dictionary attacks more successful. The focus of this scheme was towards strong password authentication.

The OSPA protocol consists of two phases, the Registration phase and the Authentication phase as follows. The notations used in the scheme are as under.

- A indicates the User.
- ID indicates User's identity.
- S indicates the Server.
- h indicates a cryptographic hash function.
- $h(m)$ means the message m is hashed once, while $h^2(m)$ means m is hashed two times
- n indicates a random nonce.
- P indicates the strong password of U .
- K indicates value stored in Smart Card.
- $+$ indicates the bitwise XOR operation.
- $//$ denotes the concatenation.
- The expression $U \rightarrow S: M$ means that user U sends the message M to S .
- (n) indicates the scheme's step number where initially $n=1$ and n increments with 1 after each step.

4.2.1 Registration Phase

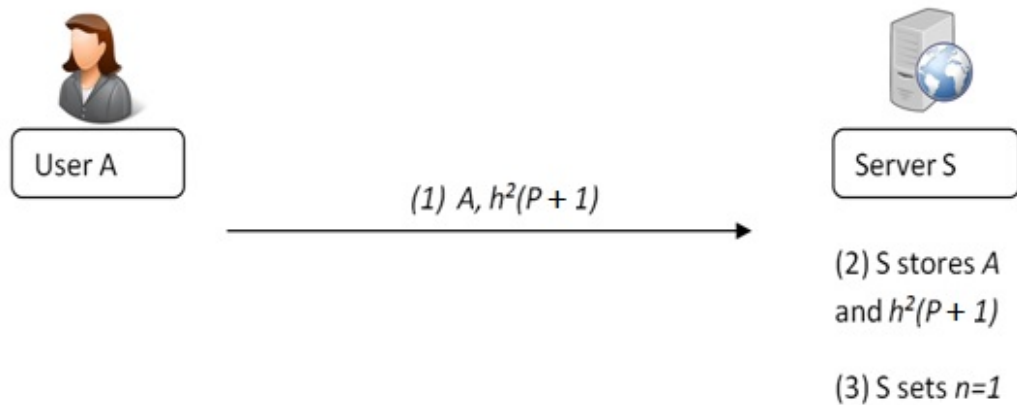


Figure 4.1: Registration Phase of OSPA Protocol

Message: A => S: A, h²(P+1)

User A calculates $h^2(P+1)$ with his password P and the initial value $n=1$, then sends the message “A, $h^2(P+1)$ ” to server S through a secure channel for registration. After receiving the registration message, server S stores A, $h^2(P+1)$ and sets $n=1$ for later authentication.

4.2.2 Authentication Phase

After registration, the i^{th} authentication procedure is described as follows, where $i \geq 1$. Figure 4.2 shows the i^{th} authentication phase of the OSPA protocol.

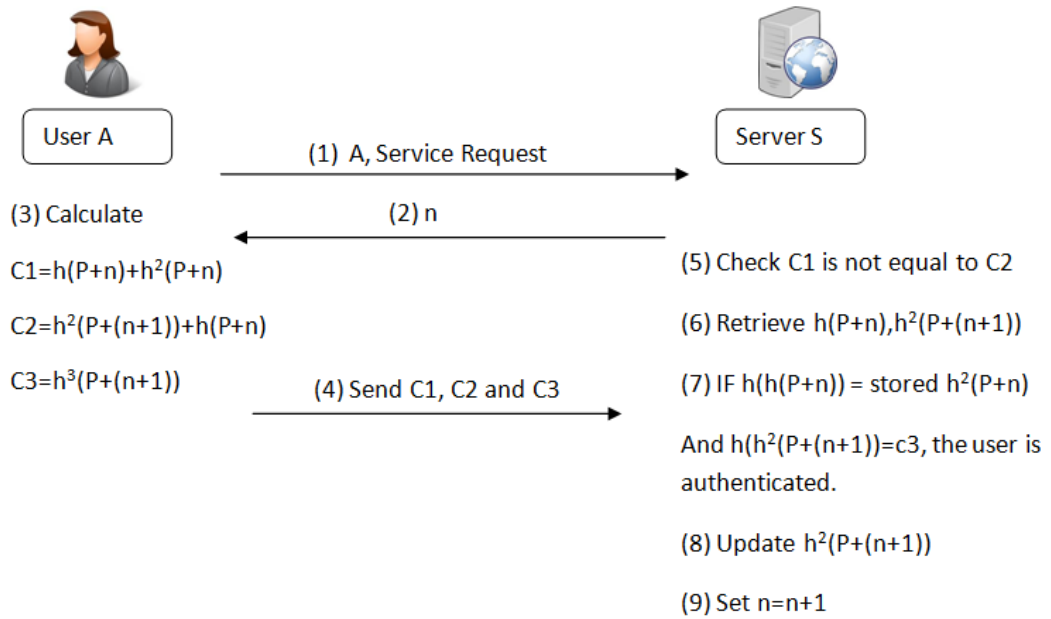


Figure 4.2: Authentication Phase of OSPA Protocol

Message 1: A => S: A, Service Request

User A issues a login service request to server S.

Message 2: S => A : n

Server S responds to A with A's i^{th} sequential number $n=i$.

Message 3: A => S: C1, C2 and C3

User A calculates three values:

(1) $C1 = h(P + n) + h^2(P + n)$. This is for the current authentication session.

(2) $C2 = h^2(P + (n + 1)) + h(P + n)$. This is for updating the next password-verifier.

(3) $c_3 = h^3(P + (n + 1))$. This is for an integrity check of updating. Then

A sends these three calculated values to server S.

After receiving the Authentication message, server S first checks whether C_1 is not equal to C_2 . If it holds, then server S retrieves $h(P+n)$ from XORing C_1 and stored value. And $h^2(P + (n + 1))$ by XORing $h(P+n)$ with C_2 .

Server S passes the authentication only if C_1 is not equal to C_2 , the value obtained from calculating $h(h(P+n))$ is equal to the stored value, and $h(h^2(P + (n + 1))) = C_3$. If the authentication is passed, server S updates stored value with $h^2(P + (n + 1))$ and sets $n=n+1$ for the next authentication session.

4.3 Enhanced- OSPA (E-OSPA)

CHEN-KU [8] and TSUJI-SHIMIZU [9] have demonstrated SV attack and Impersonation attack respectively on OSPA. Hereafter, LIN-SHEN-HWANG [10] proposed enhancements in OSPA protocol. Two new parameters were used in Enhanced-OSPA (E-OSPA). It was claimed by authors that E-OSPA can withstand against Guessing attack, Replay attack, Impersonation attack and SV attack.

The E-OSPA protocol is composed of two phases: Registration phase and Authentication phase. The notations used in the scheme are same as of OSPA.

4.3.1 Registration Phase

Registration phase consists of five steps that are as follows. The steps of registration phase have also been diagrammatically shown in Figure 4.3.

Step(1). U calculates $h^2(P \oplus N)$ and sends it with user ID to the S .

Step(2). S checks the identity of the U .

- Step(3).** S stores $h^2(P \oplus N)$.
- Step(4).** S writes $K = h^2(P \oplus N) + h(x // ID)$ on Smart Card.
- Step(5).** S issues Smart Card to the U .

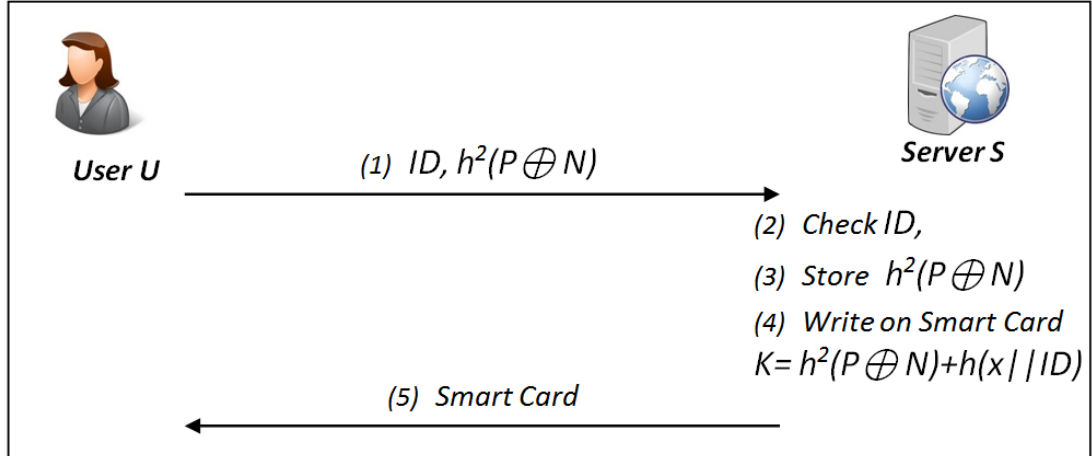


Figure 4.3: Registration Phase of E-OSPA Protocol

4.3.2 Authentication Phase

Authentication phase consists of eight steps that are as follows. The steps of authentication phase have also been diagrammatically shown in Figure 4.4.

- Step(1).** U calculates C_1, C_2 & C_3 which are $C_1 = K \oplus h^2(P \oplus N)$, $C_2 = C_1 \oplus h(P \oplus N)$ and $C_3 = h(P \oplus N) \oplus h^2(P \oplus N)^*$
- Step(2).** U sends ID, C_2 & C_3 to the S .
- Step(3).** S checks the identity of the U .
- Step(4).** S computes $C_2' = C_2 \oplus h(x // ID)$
- Step(5).** S checks whether $h(C_2')$ is equal to $h^2(P \oplus N)$.
- Step(6).** If the condition of Step5 satisfies, then S computes $h^2(P \oplus N)^* = C_3 \oplus C_2'$
- Step(7).** S authenticates U and S replaces $h^2(P \oplus N)$ with $h^2(P \oplus N)^*$.

Step(8). At user-end Smart Card replaces $h^2(P \oplus N)$ with $h^2(P \oplus N^*)$.

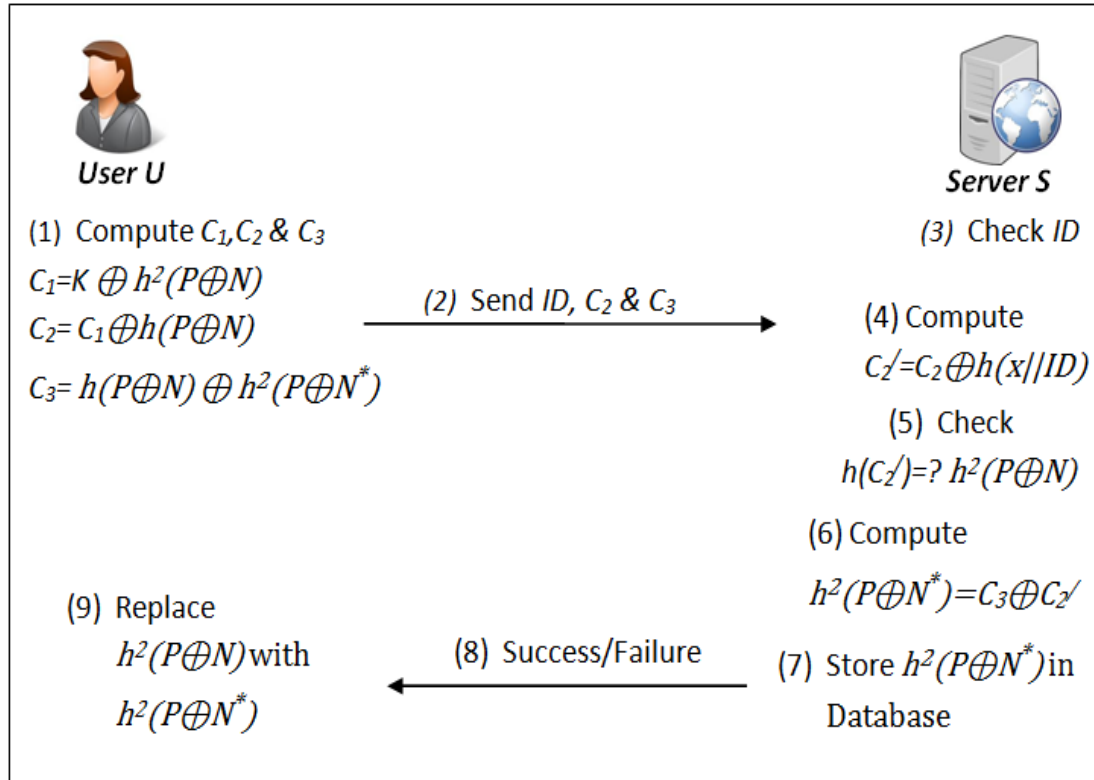


Figure 4.4: Authentication Phase of E-OSPA Protocol

4.4 W.C. Ku's Password Authentication Scheme

When KU-TSAI-CHEN [11] demonstrated replay attack and DOS attack on E-OSPA then W.C. Ku [14] proposed a more secure hash based password authentication scheme without using smart card. The major limitations of the previously proposed strong password authentication schemes were mostly because of two unsolved problems. First, if the adversary has stolen the verifier than he can impersonate the legitimate user and secondly, the integrity of next verifier being transmitted between both parties has not been well protected. So hereafter, W.C. Ku proposed newer protocol to remedy these flaws but instead of the smart card, timestamp is used in the scheme.

The scheme consists of two phases: registration phase and login phase, as described below. The notations of the scheme are same as of OSPA but below few more notations are also used in this scheme.

- n indicates total number of acceptable logins.
- T indicates the latest time User initially registers or re-registers.
- r is a random nonce set by server.
- X indicates the secret-key of S .
- K_A^T indicates the storage key of server.
- $U \leftrightarrow S: M$, means that user U sends the message M to S via secure channel.

4.4.1 Registration Phase

Registration phase consists of eight steps that are as follows. The steps are described in Figure 4.5.

- Step(1).** U sends his registration request to S .
- Step(2).** S sets T to the value of his current timestamp. If it is A 's initial registration.
- Step(3).** S sets N to 1 , or increment N by 1 i.e. $N=N+1$ if N is already set.
- Step(4).** S sends N and T to A .
- Step(5).** U computes verifier $h^2(S||P||T||N)$ and sends it to S via secure channel.
- Step(6).** S computes the storage key $K_{ID}^T = h(ID||h(x||T))$ which is only used by server. S stores K_{ID}^T in password file.
- Step(7).** S seals verifier using K_{ID}^T which is computed as:
- $$sv^N = h^2(S||P||T||N) \oplus K_{ID}^T$$

Step(8). S stores sv^N , T and N in password file.

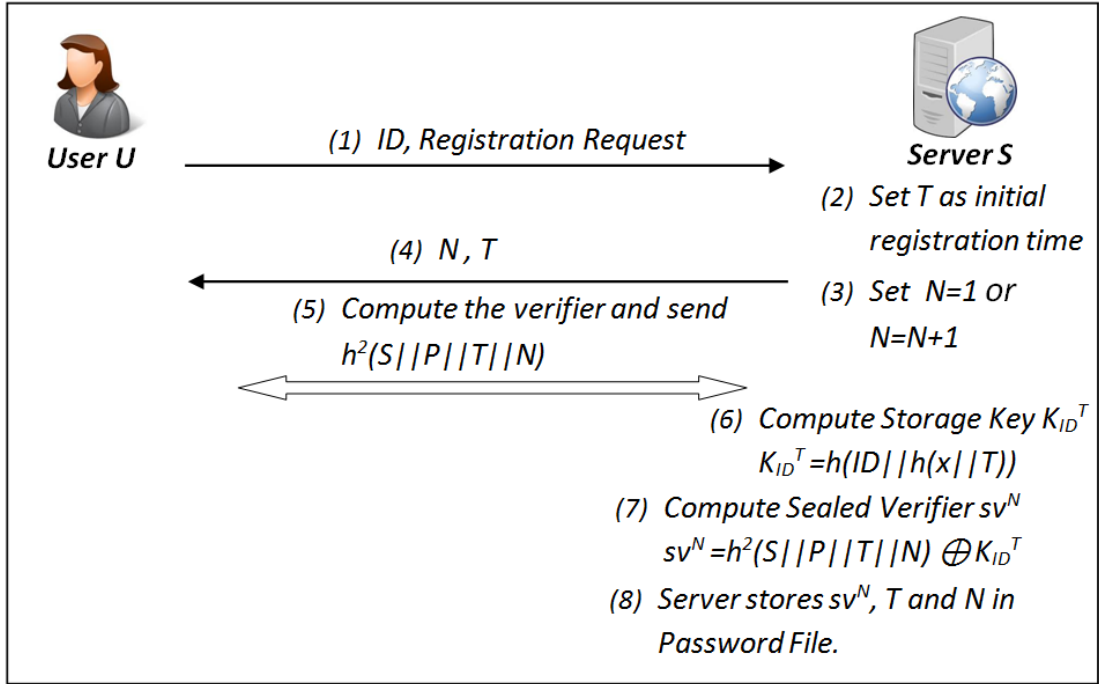


Figure 4.5: Registration Phase of W.C.Ku's Scheme

4.4.2 Login Phase

Login phase is called upon whenever U logs in S . This phase consists of fourteen steps that are as follows. Login phase has also been diagrammatically shown in Figure 4.6.

Step(1). U sends ID and login request to S .

Step(2). S checks for ID .

Step(3). S sets a random nonce r .

Step(4). S sends N, r and T to U .

Step(5). U computes C_1, C_2 & C_3 where $C_1 = h^2(S||P||T||N) \oplus h(S||P||T||N)$, $C_2 = h(S||P||T||N) \oplus h^2(S||P||T||N+1)$ and $C_3 = h(h^2(S||P||T||N+1)||r)$

Step(6). U sends C_1, C_2 & C_3 to S .

- Step(7).** S retrieves T from password file and computes $K_{ID}^T = h(ID||h(x||T))$
- Step(8).** S derives $h^2(S||P||T||N)$ by using K_{ID}^T & sv^N .
- Step(9).** S computes $U_1 = C_1 \oplus h^2(S||P||T||N) = h(S||P||T||N)$.
- Step(10).** If $Step(9)$ satisfies then S checks whether $h(U_1)$ is equal to $h^2(S||P||T||N)$.
- Step(11).** Now S computes U_2 , $U_2 = C_2 \oplus U_1 = h^2(S||P||T||N+1)$.
- Step(12).** If $Step(11)$ satisfies then S checks whether $h(U_2||r)$ is equal to C_3 .
- Step(13).** If $Step(12)$ satisfies then S sends login successful otherwise failure message to U .
- Step(14).** If the login is successful then S computes $sv^{N+1} = U_2 \oplus K_{ID}^T = h^2(S||P||T||N+1) \oplus h(ID||h(x||T))$ and replaces sv^N with sv^{N+1} & sets $N=N+1$ in the password file.

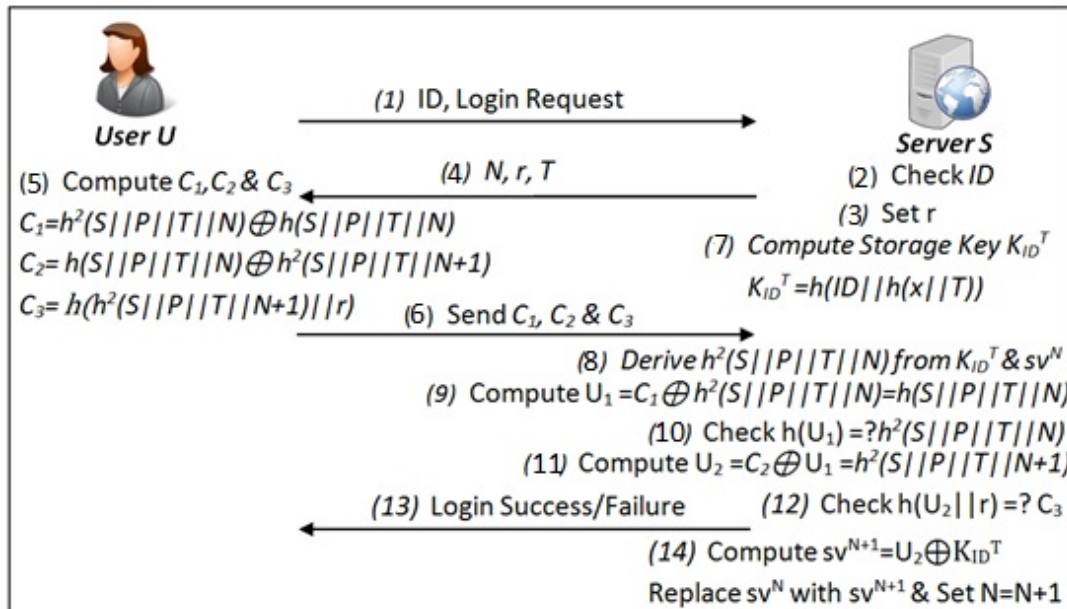


Figure 4.6: Login Phase of W. C. Ku's Scheme

In this scheme the author claimed that his scheme could withstand against replay attack, password-file compromise attack, DOS attack, impersonation attack and predictable ‘N’ attack. It was also claimed that the scheme could be easily repaired in case of collapse.

4.4.3 Kim-Koc’s Attacks on W.C. Ku’s Scheme

KIM-KOC [17] demonstrated SV attack, DOS attack, replay attack and impersonation attack on W.C. Ku’s scheme. They have devised their attack by assuming that somehow attacker got $h^2(S//P//T//N)$ and is able to block the communication. The attacker can now successfully generate C_1 , C_2 & C_3 and can authenticate to the server using replay attack and user impersonation attack. The assumption was also proved by M. Kumar [34].

Steps of the attack scenario are as under.

- Step(1).** A steals a copy of U ’s password-verifier $h^2(S//P//T//N)$.
- Step(2).** During the U ’s n th login process, A observes the communication channel, and then sees $Step(1)$ and $Step(4)$ of Login Phase of W.C. Ku’s scheme. Hereby, A knows the values of N , r and T .
- Step(3).** A captures the values of C_1 , C_2 & C_3 sent by U to S and blocks the communication channel from U to S . So, these values are not reached to S as communication line has been blocked by A .
- Step(4).** A computes $C_1 \oplus h^2(S//P//T//N) = h(S//P//T//N)$.
- Step(5).** A computes $C_2 \oplus h(S//P//T//N) = h^2(S//P//T//N+1)$
- Step(6).** So now A can derive his own C_1 , C_2 & C_3 and by impersonating U and by replaying messages, A can easily be

authenticated as the values being send to S , are correct. Hereby, the actual user U would not be able to authenticate anymore because server S will also change $N=N+I$ and sealed verifier (sv^n) after successful login of A .

4.4.4 Security Weaknesses of W. C. Ku's Scheme by M. Kumar

In 2010, M. Kumar [34] found that W.C. Ku's scheme was failed to provide session-key generation phase for secure communication and password changing phase to improve user friendliness. Moreover, the scheme does not maintain mutual authentication, Author demonstrated insider attack, parallel session attack, guessing attack, MiTM attack, SV attack, impersonation attack and DOS attack on W.C. Ku's scheme.

4.4.4.1 MiTM Attack

To apply MiTM attack the attacker must monitor and intercept the conversation over the insecure network. The attacker uses a program as an eavesdropper. In this way, the entire conversation is controlled by the attacker and the attacker must be able to monitor and intercept or modify all messages going between the two parties and pervade new ones. A MiTM attack can only be successful when the attacker can masquerade to each victim to the satisfaction of the other.

Whenever U sends his login request to S , author proves that how a malicious user A can mount *MITM* attack on W. C. Ku's hash based strong password authentication scheme in following steps.

- Step(1).** When U requests for login, attacker A intercepts the message and replaces the login request of U say L_U with his own request say L_A
- Step(2).** Attacker sends L_A to S .
- Step(3).** Server intercepts message $S \rightarrow U : r, N, T$ and replaces value with his own r_A, N_A, T_A and sends it to U .
- Step(4).** A intercepts the message $U \rightarrow S : C_1, C_2$ & C_3 and replaces it with C_1^A, C_2^A & C_3^A where $C_1^A = h^2(S||P_A||T_A||N_A) \oplus h(S||P_A||T_A||N_A)$, $C_2^A = h(S||P_A||T_A||N_A) \oplus h^2(S||P_A||T_A||N_A+1)$ and $C_3^A = h(h^2(S||P_A||T_A||N_A+1)||r_A)$
- Step(5).** A sends C_1^A, C_2^A & C_3^A to S .
- Step(6).** Server derives $h^2(S||P_A||T_A||N_A)$ from K_{ID}^T & sv^N .
- Step(7).** Server follows its complete procedure and as the data being send, satisfies all the conditions required at server end like $h(U_1) = h^2(S||P_A||T_A||N_A)$ and $h(U_2||r_A) = C_3^A$, so every check at server end would satisfy and A would be authenticated for current session.

4.4.4.2 Insider Attack

The author claimed that the scheme is not offering benefit of *scalability*, if the user U uses the same password to access other servers for ease, the insider at server S can masquerade as the user U to access other services. Thus, insider can also manipulate different components of organization's network and its security on behalf of user U . If the insider at server S is in possession of the following information.

1. Current values of N and T because these are stored in password-file.

2. U 's password-verifier $h^2(S||P||T||N)$.
3. Past records of N , r and T .
4. Values of C_1 , C_2 & C_3 .
5. Values of U_1 & U_2 .

Now, by having this set of valuable information, the insider at server can escalate an attack of his choice without knowing the related password of a valid user. The insider will be able to mount, DOS attack, SV attack, password guessing attack, impersonation attack, and replay attack etc. Thus, in W. C. Ku's scheme, the insider is a strong adversary and can do the following malicious activities.

1. Use valuable propriety information of the company's network for further exploitation.
2. Insider can place malicious spywares like Trojan Horse etc.
3. Insider can compromise availability of the system by overloading the processing or storage capacity.

4.4.4.3 Guessing Attack

As the user of the system is not free to change his/her password, therefore attacker can launch offline password guessing attack by recording the login phase and getting valuable information like N , r , T , C_1 , C_2 & C_3 .

Steps of the attack scenario are as under.

- Step(1).** Set the value N , T for the server S .
- Step(2).** Guess the password P .
- Step(3).** Compute $C_i = h^2(S||P||T||N) \oplus h(S||P||T||N)$.

Step(4). Check whether $C_i = C_{il}$ if it satisfies then it means that the attacker has managed to guess a valid password, otherwise go to *Step(2)* and set a different password P .

So, by following above given steps, attacker can successfully launch guessing attack on W.C. Ku's scheme.

4.4.5 Yang-Shen's Comments on W. C. Ku's Scheme

YANG-SHEN [36], in 2010, suggested some modifications for the improvement of W. C. Ku's scheme to resist SV attack as they have also explained the possibilities of this attack on W. C. Ku's scheme. According to the suggestions, the registration phase is as same as the original W. C. Ku's scheme except that the server would expect that user U will store nonce N and in authentication phase, S would send r and T to U and user U would have the correct value of N . Furthermore, the scheme remains same.

4.4.6 Proposed Attacks on W.C. Ku's Scheme

In this section, the vulnerabilities of W.C. Ku's scheme that have been previously found, are exploited by assuming that attacker A has got user's password verifier ($h(S//P//T//N)$) with the help of an insider. Now A can launch MiTM attack, SV attack and DOS attack by following below steps.

Step(1). Attacker A gets access between user and server as a MITM.

Step(2). A chooses some password P_A , nonce N_A , timestamp T_A and computes his own C_2^A & C_3^A .

Step(3). When U requests for login, A intercepts the message and replaces the login request of U say L_U with his own request say L_A

- Step(4).** A sends L_A to S .
- Step(5).** S intercepts message $S \rightarrow U : r, N, T$ and replaces value with his own r_A, N, T and sends it to U .
- Step(6).** A intercepts the message $U \rightarrow S : C_1, C_2$ & C_3 and replaces it with C_1, C_2^A & C_3^A where $C_1 = h^2(S//P//T//N) \oplus h(S//P//T//N)$, $C_2^A = h(S//P//T//N) \oplus h^2(S//P_A//T_A//N_A+1)$ and $C_3^A = h(h^2(S//P_A//T_A//N_A+1)//r)$
- Step(7).** A sends C_1, C_2^A & C_3^A to S .
- Step(8).** S then retrieves T from password file and computes $K_{ID}^T = h(ID//h(x//T))$
- Step(9).** Now S derives $h^2(S//P//T//N)$ by using K_{ID}^T & sv^N which is true in this scenario.
- Step(10).** S computes $U_1 = C_1 \oplus h^2(S//P//T//N) = h(S//P//T//N)$.
- Step(11).** Now $Step(9)$ will satisfy the condition so S will check whether $h(U_1)$ is equal to $h^2(S//P//T//N)$ and this condition satisfies in this scenario.
- Step(12).** Now S computes U_2 where $U_2 = C_2^A \oplus U_1 = h^2(S//P_A//T_A//N_A+1)$.
- Step(13).** Now $Step(11)$ satisfies then S checks whether $h(U_2//r_A)$ is equal to C_3 .
- Step(14).** Now $Step(12)$ satisfies then S sends login successful message to A .
- Step(15).** S computes $sv^{N+1} = U_2 \oplus K_{ID}^T = h^2(S//P_A//T_A//N_A+1) \oplus h(ID//h(x//T))$ and replaces sv^N with sv^{N+1} & sets $N=N+1$.

So, by using above steps, A can successfully authenticate with S and hereafter, as the values of password verifier and sealed verifier have been changed, so the legitimate user U would not be able to authenticate himself/herself to the S .

4.4.7 Analysis of W. C. Ku's Scheme

W. C. Ku's scheme has been analyzed with respect to our proposed framework. In Table 4.1, goals of framework have been shown whether the scheme has achieved or not. "YES" indicates that scheme has achieved the goal so far and "NO" indicates that scheme failed to achieve specified goal. The reasons have been described in the REASON column.

Table 4.1: 'Goals' Analysis of W. C. Ku's Scheme

GOAL No.	ACHIEVED YES / NO	REASONS
G1	NO	User's credentials (passwords and verifiers) are stored in plain.
G2	NO	Remote client or user can not change passwords without restraints.
G3	NO	The insiders or system administrators can reveal user's credentials.
G4	NO	Some information is transmitted in plain over the insecure network.
G5	NO	Password length is not specified.
G6	NO	The effectiveness and realistic abilities are not assured.
G7	NO	Use of wrong credentials for unauthorized login cannot be detected rapidly.
G8	NO	Key for next session is not established during authentication phase.
G9	NO	The user ID is not dynamic for every session to evade outflow of fractional information.
G10	NO	The scheme doesn't remain secure if somehow server's secret key is compromised.
G11	YES	The scheme has a lesser amount of storage and processing requirements.
G12	YES	The scheme has a reduced operating cost for transmission over the network.

In Table 4.2, benefits of framework have been shown whether the scheme has provided or not. “YES” indicates that scheme has provided the benefit so far and “NO” indicates that scheme failed to provide specified benefit. The reasons have been described in the REASON column.

Table 4.2: 'Benefits' Analysis of W. C. Ku's Scheme

BENEFIT NO.	PROVIDED YES / NO	REASON
UB1	YES	User has to remember one password only.
UB2	NO	For every account ID user has to remember each password.
UB3	YES	User has nothing to carry.
UB4	NO	User has to type password and use browser.
UB5	YES	Scheme is very simple and easy to learn.
UB6	NO	The efficiency and practical abilities are not described.
UB7	NO	Scheme failed to withstand maximum attacks and achieve goals. It will produce frequent false positive errors.
UB8	NO	Scheme is not easy reparable.
UB9	NO	User has to type password and use browser.
UB10	YES	The scheme has less cost to run.
UB11	YES	Compatible in many platforms.
UB12	YES	Compatible in any kind of end user's device.

In Table 4.3, security requirements of framework have been shown whether the scheme has fulfilled or not. “YES” indicates that scheme has fulfilled the security requirement so far and “NO” indicates that scheme failed to fulfill specified requirement. The reasons have been described in the REASON column.

Table 4.3: 'Security Requirements' Analysis of W. C. Ku's Scheme

SEC. REQ. No.	FULFILLED YES / NO	REASON
SR1	NO	It has been demonstrated in [17] and [34] that attacker can perform different attack by physical access.
SR2	NO	User impersonation attack has been performed in [17] and [34].

SR3	NO	Server spoofing attack has been performed in [17] and [34].
SR4	NO	This attack has been demonstrated in this research in section 4.4.6.
SR5	NO	This attack has been demonstrated in this research in section 4.4.6.
SR6	NO	Parallel session attack has been performed in [17].
SR7	NO	Replay attack has been performed in [34].
SR8	NO	Guessing attack has been performed in [17].
SR9	NO	This attack has been demonstrated in this research in section 4.4.6.
SR10	NO	This attack has been demonstrated in this research in section 4.4.6.
SR11	NO	As the insider attack by [34] is possible, so sealed verifier can be compromised from password-file.
SR12	NO	The Nonce 'N' can be monitored because it is being sent by server in plain.
SR13	YES	Smart card is not used in this scheme.
SR14	NO	In reference paper [26], it has been claimed that the scheme does not provide User Anonymity.
SR15	NO	Both user and server are not mutually authenticated in this scheme.
SR16	NO	As per attacks demonstrated in [17], Attacker can get storage key and sealed verifier by stealing server's secret key.
SR17	YES	No trusted party has been used
SR18	NO	Lack of State synchronization has been described in [33].

4.5 CompChall Authentication Protocol

The most popular practice used to counter online dictionary attacks on an authentication protocol is by using hash based challenge response technique. Though these techniques are the handiest but they fetch along themselves other related and hazardous threats like replay attacks, impersonation attacks and DOS attacks. There is no adequate and adoptable solution designed so far that could ultimately counter all hazards and produce arduous possibilities for an adversary to exploit and insignificant

impossibilities for genuine user to use services properly. V. Goyal et al. proposed CompChall authentication protocol [23] with claiming that their protocol can counter online dictionary attacks. However, we find that their scheme is highly vulnerable to Impersonation attack and DOS attack.

Authentication is fundamental aspect of the security, authorization and accountability of the system services. Many authentication schemes based on computational challenge response technique have been developed so far. Leslie Lamport's Hash Chain concept has been the concentrated theme of all these schemes. But till date every scheme lacks somewhere within the requirements of security, usage and installing abilities as described in previous sections.

Password based authentication schemes mainly suffer from dictionary attacks. To counter these attacks V.Goyal et al. proposed CompChall protocol to counter online dictionary attacks. Their scheme was based on hash chain mechanism to produce challenge response technique. They claimed that their protocol is perfectly stateless and it eliminates the possibility of a large number of password guesses in a small time interval by making it difficult and expensive. In this research, it has been shown that their protocol suffers from vulnerabilities to impersonation and DOS.

4.5.1 Review of CompChall Protocol

In this section, the CompChall protocol has been reviewed. The Notations used in the protocol are:

- K_{Bob} is used for secret key of the server (Bob).
- P for secret password of the user (Alice).
- 'n' is used to identify total number of unsuccessful login attempts. This parameter is only managed by server.

- ‘r’ is for the identification of 20-bits random number generated by server.
- ‘R’ is for the identification of 128-bits random number generated by server.
- MAC is the Message Authentication Code $H(H(r,R), Alice, K_{Bob}, n)$
- $H^i(x)$ is the i^{th} hash value of ‘x’.

The protocol consists of four messages in which two are exchanged without encryption and the other two involved hash computation as shown in Figure 4.7.

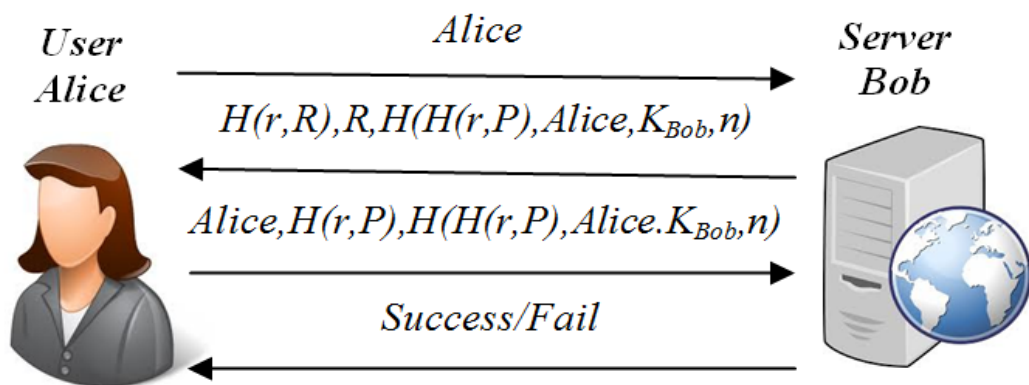


Figure 4.7: CompChall Authentication Protocol 1

Message 1: Alice

The user Alice requests for login.

Message 2: $H(r,R), R, H(H(r,P), Alice, K_{Bob}, n)$

Server generates MAC and replies with a challenge $H(r,R)$ with value of ‘R’ in plain and the message authentication code in hash.

MAC is unintelligible for user. It can only be generated by server using server's secret key K_{Bob} . The MAC is used by server to check correctness of ' r ' and the freshness of message by using ' n ' which represents number of times user could be given chance for unsuccessful authentication attempts.

The task of finding value of ' r ' for every authentication attempt increases computation and time interval for adversary to make up dictionary attack successful. The value of ' r ' doesn't change for the period of ' n ' which facilitates legitimate user to use same computation until ' n ' declines to 0.

Message 3: Alice, $H(r,P)$, $H(H(r,P), Alice.K_{Bob}, n)$

User has to find out the actual value of ' r ' by using brute force attack on $H(r,R)$ and then send its identity, hash of ' P ' appended with discovered ' r ' and received MAC. This step was made up to make the protocol stateless and secure against eavesdropping.

Message 4: Success/Fail

The server compares the values of received ' r ' and ' P ' by using MAC. If the values are correct, the user would be authenticated otherwise server will decrement the value of ' n ' by 1 and the protocol will start all over again with fresh ' r '.

Lamport's hashes were employed in a variant of above protocol to avoid the use of SSL layer protection and storing password P in plaintext at server. The modified messages of this variant is shown in Figure 4.8.

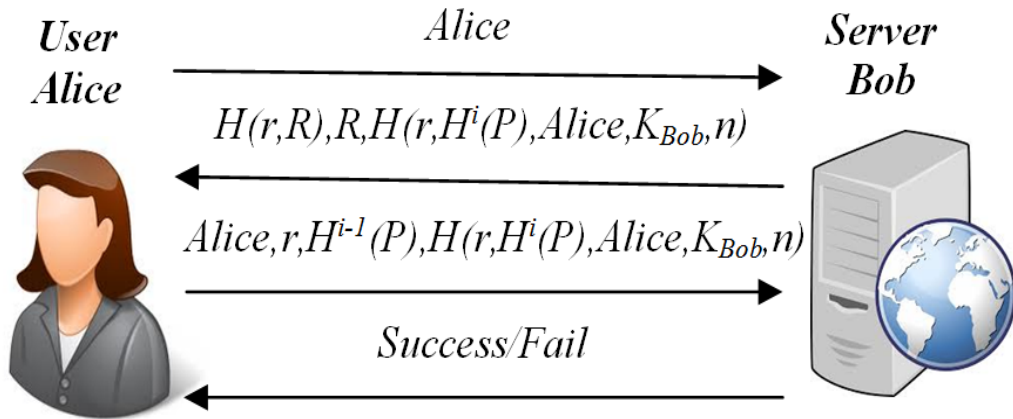


Figure 4.8: CompChall Authentication Protocol 2

Server stores $H^i(P)$ which is i^{th} hash of P . User is being required to supply $H^{i-1}(P)$ as a password. Server has to calculate $H^{i-1}(P)$ and compare it with the received one and respond with success or failure message. After successful login attempt, the stored $H^i(P)$ is replaced by the supplied $H^{i-1}(P)$. Thus next time the user has to supply $H^{i-2}(P)$. This process continues for i successful login attempts after which re-initialization of protocol by choosing different password after i successful login attempts would be required. The 3rd message is modified in such a way that user has to send ' r ' and $H^{i-1}(P)$ in plain.

4.5.2 Proposed Attacks on CompChall Protocol

Some new notations used for the demonstration of attack on CompChall protocol are:

- K_{Oracle} is used for secret key of the attacker (Oracle).
- P^* Replica secret password for the user (Alice).
- ' n^* ' indicates total number of unsuccessful login attempts. This parameter is managed by an attacker.

- ‘ r^* ’ is for the identification of 20-bits random number generated by attacker.
- ‘ R ’ is for the identification of 128-bits random number generated by attacker.
- MAC^* is $H(r^*, H^{i-1}(P^*), Alice, K_{Oracle}, n^*)$.
- $H^i(x)$ is the i^{th} hash value of ‘ x ’.

The problem found in this protocol is that the server is not authenticating itself to user by any means. As we can see that *Alice* is sending the next session key i.e. $H^{i-1}(P)$ and ‘ r ’ in plain without any further hashing or encryption. Both values are sent disjointedly. Attacker (*Oracle*) can sniff these values over insecure network. However both these values are worthless for the *Oracle* for ongoing session but these values can be used for next session by using phishing and impersonation attack where *Oracle* could pretend to be an actual server. Moreover, after successful authentication both parties are dropping their previous hash values and considering $H^{i-1}(P)$ to be a new session key. As actual *MAC* is unintelligible for *Alice*, so if *Oracle* generates his own MAC^* which could be like $H(r^*, H^{i-1}(P^*), Alice, K_{Oracle}, n)$, would be considered as actual *MAC* by *Alice*. Now *Oracle* can deny the services for *Alice* and compromise user’s confidentiality using phishing attacks. The process of attack has been demonstrated in Figure 4.9.

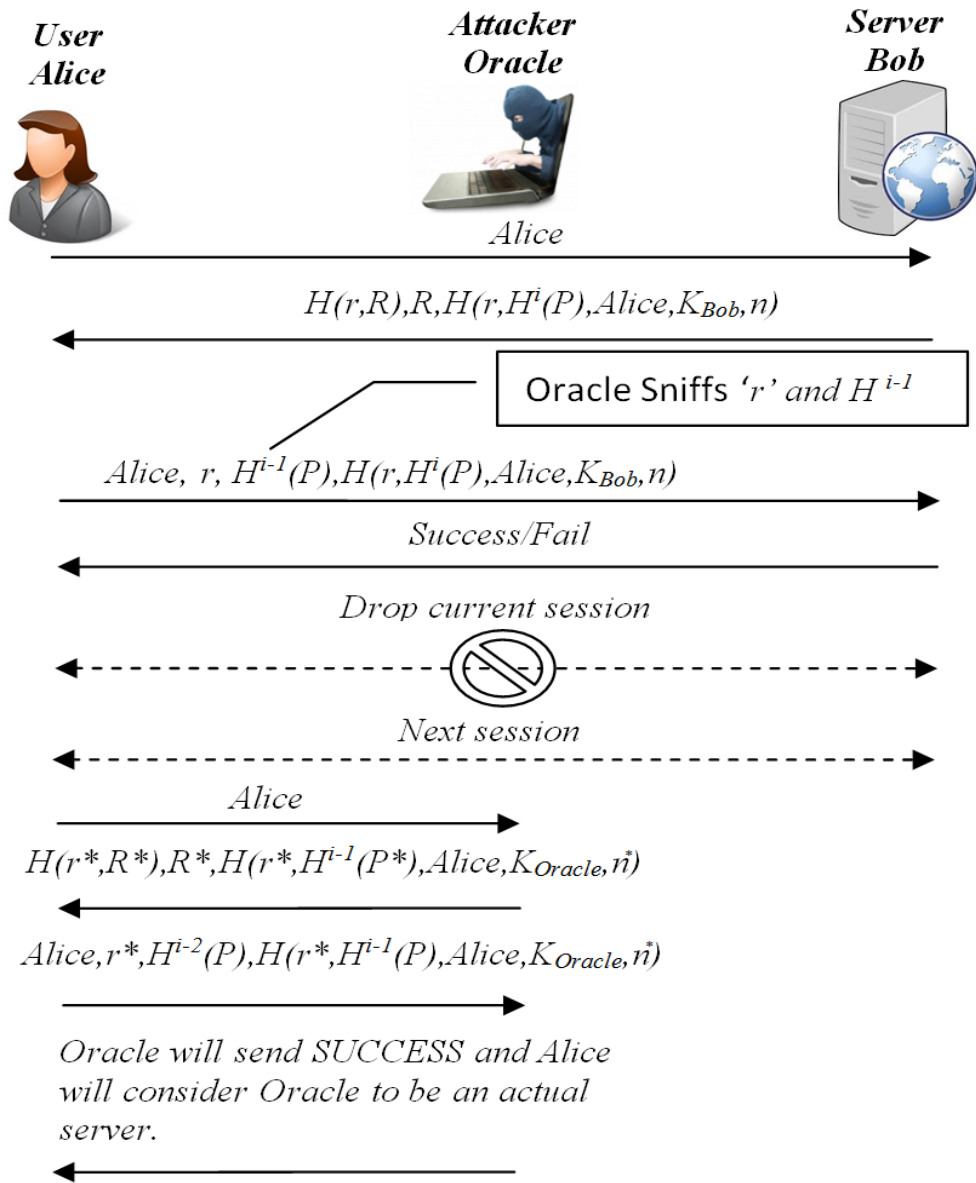


Figure 4.9: Attacks on CompChall Authentication Protocol

4.6 Conclusions

Hash chain based password authentication schemes have been reviewed and analysis and comparisons between them have been presented so far. OSPA, E-OSPA and W. C. Ku's schemes are the most preferred for most of the platforms. W. C. Ku's scheme has been shown vulnerable and then improved many times in last twelve years. A combination of attacks has also been demonstrated in W. C. Ku's scheme and CompChall protocol so far.

PROPOSED SCHEME

5.1 Introduction

Many authentication schemes based on lamport's hash chain [01] method have been proposed so far, however, none of them has successfully brought suitable protocol to fulfill highest level of desired criteria of our given framework which is being discussed in Chapter 3. All of them are still considered vulnerable to different types of attacks and lacking from different goals to achieve and provide benefits for different scenarios. OSPA protocol, E-OSPA protocol and W. C. Ku's scheme has been remained center of attention for so many years, improved and then again broken by so many times [08] [09] [10] [11] [12] [15] [26] [27] [38]. But still hash based password authentications schemes are preferred the most for most of the scenarios and remained unfasten research area, because a strong password authentication is still needed to the world. In this chapter, a new scheme has been proposed named Hash Chain based Strong Password Authentication Scheme (HC-SPAS).

5.2 Hash Chain based Strong Password Authentication Scheme

In 2004, Ku proposed a more secure hash based password authentication scheme without using smart card with the consideration of the need of a highly efficient password authentication scheme having reduced storage, processing and transmission overhead. The protocol has been well explained in section 4.4.

Ku claimed that his proposed scheme can resist to the several attacks including replay attack, password-file compromise or SV attack, DOS attack and insider attack, but in 2005, KIM-KOC [17] demonstrated SV attack, DOS attack, replay attack and

impersonation attack on W.C. Ku's scheme. Furthermore in 2010, M. Kumar [34] found some flaws on ku's scheme. Moreover, the scheme is not user friendly in terms of password changing phase. Author demonstrated insider attack, parallel session attack, guessing attack, MiTM attack, SV attack, impersonation attack and DOS attack on W.C. Ku's scheme.

In addition to previously found vulnerabilities in Ku's scheme, YANG-SHEN [35] further demonstrated a simple MiTM attack in 2010. Subsequently, authors have presented some improvements and extended W.C. Ku's scheme in form of a new protocol named as Strong Password Authentication Scheme (SPAS). The authors claimed that SPAS protocol can withstand the SV attack. Recently, Xu et al. [40] proposed password based authentication scheme based on geometric hashing function without using smart card.

As in section 4.4.6, the proposed attacks have been demonstrated and explained that the improved scheme is still need to be refined. In this section, HC-SPAS (Hash Chain based Strong Password Authentication Scheme) has been proposed and demonstrated that the proposed scheme is more efficient and robust than previously presented schemes according to the desired criteria. Our scheme consists of three phases: registration phase, login phase and password change phase.

5.3 Registration Phase

This phase is invoked whenever user U initially registers or re-registers to server S . A secure channel has been only assumed for the initial message between U and S . The process of registration phase is:

- Step(1).** U chooses N_i & $m_i \neq 0$ or 1 , where N_i represents nonce and m_i indicates number of iterations U can use for login using same password. Now U sends his identity denoted by ID , registration request, N_i & m_i to S via secure channel. S stores ID, N_i & m_i in password-file and sets T to the value of his current timestamp.
- Step(2).** $S \rightarrow U : H^{m_i}(N_i - 1) \oplus T$. Here H^{m_i} means m_i^{th} hash and \oplus denotes XOR operator. Using N_i & m_i , U computes $T = H^{m_i}(N_i - 1) \oplus H^{m_i}(N_i - 1) \oplus T$. U chooses password P and computes $H^2(S || H^{m_i}(P) || T || N_i) = Z$. U computes $B = Z \oplus H^{m_i-1}(N_i - 1) \oplus T$ and $U \rightarrow S : B$.
- Step(3).** S computes $H^{m_i-1}(N_i - 1) \oplus T = Q$ and gets Z by computing $Q \oplus B$. Using x as a secret key of S , S computes storage-key $K_{ID}^T = H(ID || H(x || T))$. S computes sealed-verifier for password-file $SV_S^{N_i} = Z \oplus K_{ID}^T$ and sealed-verifier for smart-card $SV_U^{N_i} = Z \oplus K_{ID}^T$. S computes $N_{i+1} = N_i + 1$ and $m_{i+1} = m_i - 1$.

Step(4). S writes $SV_U^{N_i}, N_i, N_{i+1}, m_i, m_{i+1}$ and T in smart-card and writes $ID, SV_S^{N_i}, K_{ID}^T, N_i, N_{i+1}, m_i, m_{i+1}$ and T on password-file. S issues smart-card to U .

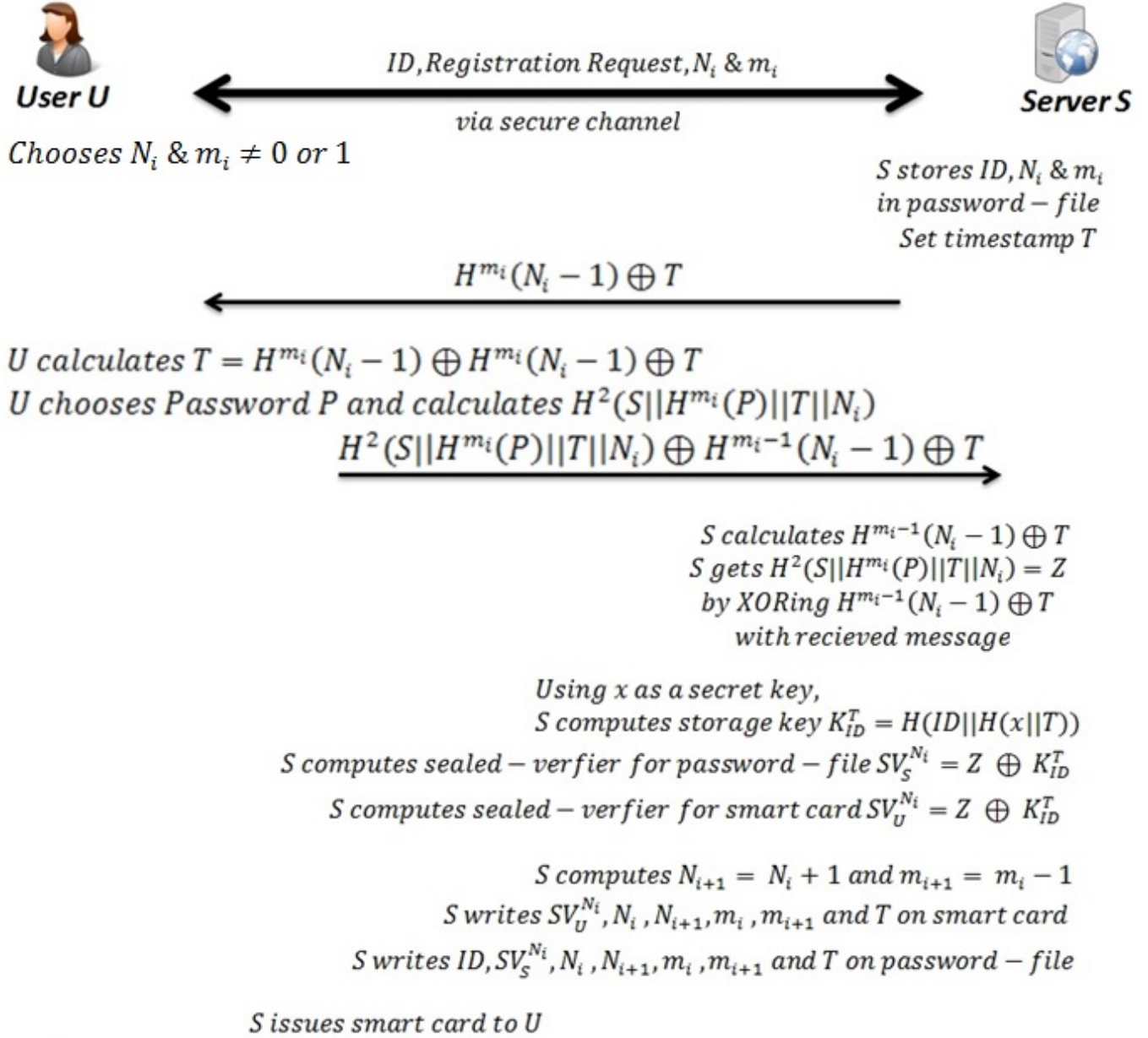


Figure 5.1: Registration Phase of Proposed Scheme

5.4 Login Phase

This phase is invoked whenever U logs in S. The steps of this phase are described below. Notation $\text{Attribute1} \leftarrow \text{Attribute2}$ means that value of Attribute2 is set to Attribute1.

- Step(1).** $U \rightarrow S : ID$ and login request. S checks for user's ID and m_i in password-file. If $m_i = 0$ or 1, then S will ask U to first change the password and password change phase will be invoked, otherwise S computes $H^{m_{i+1}}(N_{i+1})$.
- Step(2).** $S \rightarrow U : H^{m_{i+1}}(N_{i+1}) \oplus K_{ID}^T$. U enters password P and smart-card computes $K_{ID}^T = H^{m_{i+1}}(N_{i+1}) \oplus K_{ID}^T \oplus H^{m_{i+1}}(N_{i+1})$.
- Step(3).** Smart-card computes $H(H(S||H^{m_i}(P)||T||N_i)) = H(Y) \approx Z$. Smart-card checks for the condition $H(Y) \oplus K_{ID}^T =? SV_U^{N_i}$.
- Step(4).** If condition satisfies then smart-card computes $X = H^2(S||H^{m_{i+1}}(P)||T||N_{i+1})$.
- Step(5).** Smart-card computes $H^{m_{i+1}-1}(N_{i+1}) = R$. Then smart-card computes $C_1 = H(Y) \oplus Y$, $C_2 = Y \oplus X$ and $C_3 = H(X \oplus K_{ID}^T)$. $U \rightarrow S : \{R \oplus C_1\}, C_2$ and C_3 to S .
- Step(6).** S computes $C_1 = \{R \oplus C_1\} \oplus H^{m_{i+1}-1}(N_{i+1})$. To get Z which is equivalent to $H(Y)$, S computes $SV_S^{N_i} \oplus K_{ID}^T = Z$.
- Step(7).** S computes $U_1 = C_1 \oplus Z$ which is equal to $H(S||H^{m_i}(P)||T||N_i)$ and checks for condition $H(U_1) =? Z$.

Step(8). If the condition satisfies then S computes $U_2 = U_1 \oplus C_2 = H^2(S||H^{m_{i+1}}(P)||T||N_{i+1}) \approx X$. S checks for condition $H(U_2 \oplus K_{ID}^T) =? C_3$ and if satisfies then S sends login successful message to U otherwise sends failure message to U and does not change any state.

Step(9). Now if the login is successful then S computes $SV_S^{N_{i+1}} = U_2 \oplus K_{ID}^T$. S further sets values of $SV_S^{N_i} \Leftarrow SV_S^{N_{i+1}}, N_i \Leftarrow N_{i+1}$ and $m_i \Leftarrow m_{i+1}$. S computes values of N_{i+1} and m_{i+1} by calculating $N_{i+1} = N_i + 1$ and $m_{i+1} = m_i - 1$. S updates values of $SV_S^{N_i}, N_i, m_i, N_{i+1}$ and m_{i+1} in password-file.

Step(10). Meanwhile, at the user end, smart-card computes $SV_U^{N_{i+1}} = X \oplus K_{ID}^T$ and sets $SV_U^{N_i} \Leftarrow SV_U^{N_{i+1}}, N_i \Leftarrow N_{i+1}$ and $m_i \Leftarrow m_{i+1}$. Smart-card computes new values for N_{i+1} and m_{i+1} by calculating $N_{i+1} = N_i + 1$ and $m_{i+1} = m_i - 1$. Smart-card updates values of $SV_S^{N_i}, N_i, m_i, N_{i+1}$ and m_{i+1} .



ID, Login Request

$H^{m_{i+1}}(N_{i+1}) \oplus K_{ID}^T$

U enters password P

Using x as a secret key,

S computes storage key $K_{ID}^T = H(ID||H(x||T))$

Smart card calculates $H^{m_{i+1}}(N_{i+1}) \oplus K_{ID}^T \oplus H^{m_{i+1}}(N_{i+1}) = K_{ID}^T$

Calculates $H^2(S||H^{m_{i+1}}(P)||T||N_{i+1}) = X$

Calculates $H(H(S||H^{m_i}(P)||T||N_i)) = H(Y) \approx Z$ at server – end

Checks $H(Y) \oplus K_{ID}^T =? SV_U^{N_i}$

Calculates $H^{m_{i+1}-1}(N_{i+1})$

Smart card calculates

$$C_1 = H(Y) \oplus Y$$

$$C_2 = Y \oplus X$$

$$C_3 = H(X \oplus K_{ID}^T)$$

Sends $\{H^{m_{i+1}-1}(N_{i+1}) \oplus C_1\}, C_2$ and C_3

S calculates $C_1 = H^{m_{i+1}-1}(N_{i+1}) \oplus \{H^{m_{i+1}-1}(N_{i+1}) \oplus C_1\}$

S Calculates $SV_S^{N_i} \oplus K_{ID}^T = Z \approx H(Y)$ at user – end

Calculates $U_1 = C_1 \oplus Z = H(S||H^{m_i}(P)||T||N_i)$

Checks $H(U_1) =? Z$

Calculates $U_2 = U_1 \oplus C_2 = H^2(S||H^{m_{i+1}}(P)||T||N_{i+1}) \approx X$ at user – end

Checks $H(U_2 \oplus K_{ID}^T) =? C_3$

Login Successful/Failure

Smart card calculates $SV_U^{N_{i+1}} = X \oplus K_{ID}^T$

S calculates $SV_S^{N_{i+1}} = U_2 \oplus K_{ID}^T$

Smart card sets values $SV_U^{N_i} \leftarrow SV_U^{N_{i+1}}, N_i \leftarrow N_{i+1}$ and $m_i \leftarrow m_{i+1}$

Smart card computes new values of N_{i+1} and m_{i+1}

$N_{i+1} = N_i + 1$ and $m_{i+1} = m_i - 1$ S sets values $SV_S^{N_i} \leftarrow SV_S^{N_{i+1}}, N_i \leftarrow N_{i+1}$ and $m_i \leftarrow m_{i+1}$

Smart card changes values of $SV_S^{N_i}, N_i, m_i, N_{i+1}$ and m_{i+1}

S computes new values of N_{i+1} and m_{i+1}

$N_{i+1} = N_i + 1$ and $m_{i+1} = m_i - 1$

S writes values of $SV_S^{N_i}, N_i, m_i, N_{i+1}$ and m_{i+1} in password – file

Figure 5.2: Login Phase of Proposed Scheme

5.5 Password Change Phase

Password change phase is used whenever U wants to change his/her password or m_i reaches to 0 or 1. The steps for the login phase are associated with steps of login phase except from below few changes.

- Step(1).** $U \rightarrow S : ID$ and Password change request. S checks for user's ID in password-file and computes $H^{m_{i+1}}(N_{i+1})$. $S \rightarrow U :$
 $H^{m_{i+1}}(N_{i+1}) \oplus K_{ID}^T$.
- Step(2).** *Step(2)* and *Step(3)* of login phase are followed in this step.
- Step(3).** System asks U to enter new password P' . Smart-card computes
 $X = H^2(S || H^{m_{i+1}}(P') || T || N_{i+1})$.
- Step(4).** *Step(5)*, *Step(6)* and *Step(7)* of login phase are followed in this step.
- Step(5).** If the condition satisfies then S checks for condition $U_2 = U_1 \oplus C_2$ which is equal is $H^2(S || H^{m_{i+1}}(P') || T || N_{i+1}) \approx X$ of U 's. If the conditions satisfy then S checks for condition $H(U_2 \oplus K_{ID}^T) =? C_3$ and if satisfies then $S \rightarrow U : login$ and password changed successfully, otherwise sends failure message to U and does not change any state. Afterwards, *Step(9)* and *Step(10)* of login phase are followed for state synchronization.



ID, Password Change Request



Check ID

*Using x as a secret key,
S computes storage key $K_{ID}^T = H(ID||H(x||T))$*

$H^{m_{i+1}}(N_{i+1}) \oplus K_{ID}^T$

U enters prev – password P

Smart card calculates $H^{m_{i+1}}(N_{i+1}) \oplus K_{ID}^T \oplus H^{m_{i+1}}(N_{i+1}) = K_{ID}^T$

Calculates $H(H(S||H^{m_i}(P)||T||N_i)) = H(Y) \approx Z$ at server – end

Checks $H(Y) \oplus K_{ID}^T =? SV_U^{N_i}$

System asks U to enter new password P'

Calculates $H^2(S||H^{m_{i+1}}(P')||T||N_{i+1}) = X$

Calculates $H^{m_{i+1}-1}(N_{i+1})$

Smart card calculates

$$C_1 = H(Y) \oplus Y$$

$$C_2 = Y \oplus X$$

$$C_3 = H(X \oplus K_{ID}^T)$$

Sends $\{H^{m_{i+1}-1}(N_{i+1}) \oplus C_1\}, C_2$ and C_3

S calculates $C_1 = H^{m_{i+1}-1}(N_{i+1}) \oplus \{H^{m_{i+1}-1}(N_{i+1}) \oplus C_1\}$

S Calculates $SV_S^{N_i} \oplus K_{ID}^T = Z \approx H(Y)$ at user – end

Calculates $U_1 = C_1 \oplus Z = H(S||H^{m_i}(P)||T||N_i)$

Checks $H(U_1) =? Z$

Calculates $U_2 = U_1 \oplus C_2 = H^2(S||H^{m_{i+1}}(P')||T||N_{i+1}) \approx X$ at user – end

Checks $H(U_2 \oplus K_{ID}^T) =? C_3$

Login Successful/Failure

Smart card calculates $SV_U^{N_{i+1}} = X \oplus K_{ID}^T$

Smart card sets values $SV_U^{N_i} \leftarrow SV_U^{N_{i+1}}, N_i \leftarrow N_{i+1}$ and $m_i \leftarrow m_{i+1}$

Smart card computes new values of N_{i+1} and m_{i+1}

$$N_{i+1} = N_i + 1 \text{ and } m_{i+1} = m_i - 1$$

Smart card changes values of $SV_S^{N_i}, N_i, m_i, N_{i+1}$ and m_{i+1}

S calculates $SV_S^{N_{i+1}} = U_2 \oplus K_{ID}^T$

S sets values $SV_S^{N_i} \leftarrow SV_S^{N_{i+1}}, N_i \leftarrow N_{i+1}$ and $m_i \leftarrow m_{i+1}$

S computes new values of N_{i+1} and m_{i+1}

$$N_{i+1} = N_i + 1 \text{ and } m_{i+1} = m_i - 1$$

S writes values of $SV_S^{N_i}, N_i, m_i, N_{i+1}$ and m_{i+1} in password – file

Figure 5.3: Password Change Phase of Proposed Scheme

5.6 Conclusions

None of the schemes introduced so far have passed through the threshold of the proposed framework. But the proposed HC-SPAS can pass through the defined threshold successfully as proved in chapter 6. The proposed scheme has been organized in such a way that vulnerabilities being exploited in previously introduced schemes have been rectified and countered. The methods and related approaches used for attacks proposed in W. C. Ku's scheme, cannot be execute on the proposed scheme. There are three phases of proposed HC-SPAS scheme. Each phase is self-governing in nature and can be implemented in any platform.

FRAMEWORK SCRUTINY OF PROPOSED SCHEME

6.1 Introduction

In this chapter, proposed scheme (HC-SPAS) has been analyzed and compared with previously presented schemes according to desired criteria of the proposed framework. According to the analysis shown in this chapter, it has been proved that HC-SPAS scheme is far better than previously introduced schemes because of better usability, scalability, and deployability and security countermeasures taken in this scheme.

6.2 Analysis and Comparison of Proposed Scheme

This section has been divided into two parts i.e. analysis of HC-SPAS according to the framework, and comparison between HC-SPAS and W.C. Ku's scheme.

6.2.1 Analysis and Framework Scrutiny of HC-SPAS

Framework scrutiny of proposed HC-SPAS has been presented below in three parts.

6.2.1.1 Goals

- Most of the goals of framework are achieved in proposed framework as described below.

- The passwords, storage keys and verifiers are stored in the system but in the form of hash chains.
- Intruder cannot expose user's credentials as in every session user has to transmit next verifier of the hash chain. The verifiers at server are stored in hash. Intruder cannot spoof user or server as because the storage keys and password verifiers are in hash form and selected every time from a defined and shared hash chain.
- There is a complete and separate phase proposed for password changing. The verifiers are not transmitted in plain on the network.
- Password of suitable length is enough for the security of scheme because the password has been converted into hash form and transmitted as one time password after creating hashed password chains.
- The proposed scheme is robust and practical.
- If an intruder tries to pass wrong credentials during login phase, server will detect the illegitimate user because the algorithm has been designed in such a way that server expects a secret from legitimate user in every step. Server detects wrong credentials in every step and blocks illegitimate user.
- Key for next session is established during authentication phase in the form of hash chain. The next key from chain has been calculated and verified at both ends.

- No partials information has been transmitted in the flow, as every message has been hashed except the first message of registration phase which has to be transferred via secure channel.
- HC-SPAS remain secure if somehow server's secret key is compromised.
- HC-SPAS have a lesser amount of storage and processing requirements.
- HC-SPAS have a reduced operating cost for transmission over the network.

6.2.1.2 Benefits

The proposed scheme has provided maximum level of usability and deployability benefits of the proposed framework. The justifications of the claim have been described below.

- Users don't need to memorize more than passwords which is needed in at first step of each phase.
- Using the scheme for maximum number of accounts does not raise the burden on the user. User can even use same password though it's not recommended.
- Proposed scheme needs user to carry smart card. Proposed scheme doesn't provide this benefit.
- User just need to enter password, more than that every other process is followed by smart card from the user's end.
- Users can figure out steps of scheme and be trained without too much trouble, and then easily remember how to use it.

- The time required for establishing session between user and server is practical and short.
- Server detects and rejects irregular flow of steps or values at start of every step. So server doesn't produce errors.
- A user can easily recover the ability to authenticate if password is forgotten by launching password change phase. If the smart card is lost. Server has to block lost smart card and regenerate new smart card for the user.
- Any disabled person can use this scheme for authentication.
- The total price per user of the scheme, adding up the expenses at both ends is reasonable.
- Scheme is compatible with both text based password system at server end and application at user's end like browser.

6.2.1.3 Security Requirements

Except from first message of registration phase of HC-SPAS, which has to be transmitted via secure line, every other message of the scheme can be transmitted in common communication line. If our proposed scheme is used in web based applications using transport layer security enabled, the scheme shall be superlative in results. Attacker cannot impersonate user or spoof server because scheme is offering mutual authentication and every message has been hashed using hash chain method. Moreover, attacker can neither get or replay values of stored-verifier, nonce and storage-key nor create responses for the challenges because every next challenge is appended and chained with the expected response. States are always kept synchronized between both parties and attacker cannot start parallel session with both

the server and user at the same time because of hash chains being used and verifying by responses of each other's challenges. User can change password freely and system will ask to change password whenever $m_i = 0$ or 1.

It is not computationally feasible for an attacker to guess any of the parameter's value being used by server or client. If somehow attacker is able to compromise the verifier, he or she cannot generate the response of the challenge as none of the content of any message is being repeated. Storage key has been specifically used for secure storage of data at both ends and user's and server's anonymity has been maintained throughout the scheme. The previously generated passwords or verifiers in the password-file or smart-card are secure even if the system's secret key or smart-card has been public by any means.

6.2.2 Comparison between HC-SPAS and W. C. Ku's Scheme

The comparison between W. C. Ku's scheme and our proposed scheme HC-SPAS has been shown according to given framework in Table.6.1.

In the table 6.1, "Y" indicates that scheme fulfills the desired criteria of framework. "N" indicates that scheme does not fulfill the desired criteria of framework. And "#" sign indicates that scheme fulfills the desired criteria if the assumptions are executed. Both W. C. Ku's scheme and proposed scheme is compared to each other. Clearly, proposed scheme has far better results than previously presented scheme like W. C. Ku's scheme, as shown in the table.

Table 6.1: Analysis and Comparisons

Goals												
G	1	2	3	4	5	6	7	8	9	10	11	12
<i>Ku's</i>	N ^a	N	N	N	N	N	N	N	N	N	Y	Y
<i>HC-SPAS</i>	Y ^b	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Benefits												
B	<i>Usability(UB)</i>								<i>Deployability(DB)</i>			
	1	2	3	4	5	6	7	8	1	2	3	4
<i>Ku's</i>	Y	N	Y	N	Y	N	N	N	N	Y	Y	Y
<i>HC-SPAS</i>	Y	Y	N	N	Y	Y	Y	Y	Y	Y	Y	Y
Security Requirements												
SR	1	2	3	4	5	6	7	8	9	10	11	12
<i>Ku's</i>	N	N	N	N	N	N	N	N	N	N	N	N
<i>HC-SPAS</i>	# ^c	Y	Y	#	Y	Y	Y	Y	Y	Y	Y	Y
SR				13	14	15	16	17	18			
<i>Ku's</i>				Y	N	N	N	Y	N			
<i>HC-SPAS</i>				Y	Y	Y	Y	Y	Y			

^a Y indicates that scheme fulfills the desired criteria of framework.

^b N indicates that scheme does not fulfill the desired criteria of framework.

^c # indicates that scheme fulfills the desired criteria if the assumptions are executed.

6.3 Conclusions

It is very well proved now that the proposed scheme can easily pass through the threshold defined in proposed framework. Compared to the previously introduced schemes, HC-SPAS have far better results. W. C. Ku's scheme was center of attention so far but HC-SPAS have clearly more better outcomes than W. C. Ku's scheme according to the framework.

CONCLUSIONS AND FUTURE WORK

7.1 Introduction

In this chapter, a brief conclusion of the thesis and future possible work is discussed. The conclusion also consists of results and observations regarding the proposed framework and scheme. And in the future work section, some suggestions are put forward for researchers to mature the presented work in further extent.

7.2 Conclusions

Hash chain based password authentication schemes have been remained main focus in this research. Start from the beginning when the hash chains method was introduced by Lamport in 1981, many schemes have been presented. Every scheme was the improvement of previously presented scheme. OSPA, E-OSPA and W. C. Ku's schemes were center of attention and are reviewed in details in this research. It has been learnt after studying several researches and appraisals that hash chain based password authentication schemes are the most preferred due to their usability, deployability benefits and security countermeasure techniques.

A framework has been designed consists of goals, benefits, security requirements and a flow diagram for the assessment of an ideal hash chain based password authentication schemes. The framework describes a threshold that depends on the platform where scheme has to be implemented. Every scheme must pass through this predefined threshold for deployment.

W.C. Ku's scheme has been reviewed and showed that how an attacker can carry out MiTM attack, SV attack and DOS attack on his protocol. Afterwards, an enhanced and efficient hash chain based strong password authentication scheme has been presented and finally analyzed and compared both the schemes according to the desired criteria of the framework and proved that the proposed scheme is far better than other hash based authentication schemes. The proposed scheme can be used in many technologies and applications like web applications and networks.

According to the security and performance analysis of HC-SPAS with respect to the proposed framework in chapter 6, it is sure that the proposed scheme does not only keep the security features claimed by previously presented schemes, but also provide the security reassurance under every factor described in the framework which has been designed with deep concern towards security after analysis of different researches. Therefore, the proposed scheme is much more improved now and is more appropriate for real-life cryptographic submissions than previously presented works.

7.3 Future Work

The proposed scheme is effective to countermeasure the main attacks demonstrated on W. C. Ku's scheme so far. In this section, some research topics for future works have been addressed.

In the previously presented schemes and proposed schemes, almost all registration phases for password based authentications are performed through a secure channel. In this research, it has been countered till the first message of the registration phase. Other than first message, every other message can be passed through insecure channel. For example, when a new user wants to access a resource or request a service, user must follow through registration phase over secure channel. But, keeping

in view of today's internet and mobile communications, it's an open environment for everyone including intruders. Most of the applications are not supplying a secure channel to register new user. Thus, in the future, an efficient remote user registration before authentication over insecure channel is more of the concern.

In this thesis, proposed scheme can countermeasure and prevent several attacks. Unfortunately, the proposed scheme cannot completely provide security against physical observations especially registration phase, DOS attack, and insider attacks. In case of synchronized attacks, the proposed scheme may break. For example, if the insider attack occurs and the smart card has been stolen, the malicious insider can guess the password and generate verifier using smart card. Malicious user would require finding out the nonce being used in the proposed scheme i.e. "N" and "m". If somehow, insider gets able to find out nonce, he or she can generate DOS attack on the system, afterwards, malicious user would be blocked. In future, it is needed to propose a solution to counter these problems.

BIBLIOGRAPHY

- [1] L. Lamport, "Password authentication with insecure communications," *Commun. ACM*, vol.24, no.11, pp.770-772, 1981.
- [2] A. Shimizu, "A dynamic password authentication method by one-way function," *IEICE Transactions and Communications*, Vol.J73-D-I, no.7, pp.630-636, July 1990.
- [3] N. Haller, "The S/Key TM one-time password system," *Proc. Internet Society Symposium on Network and Distributed System Security*, pp.151-158, 1994.
- [4] L.Chen and C.J. Mitchell, "Comments on S/Key user authentication system," *ACM Operating System Review*, vol. 30, No.4, pp. 12-16. 1996.
- [5] A. Shimizu, T. Horioka, and H. Inagaki, "A password authentication method for contents communication on the internet," *IEICE Transactions and Communications*, vol.E81-B,no.8,pp.1666-1673, Aug. 1998.
- [6] M. Sandirigama, A. Shimizu, and M.T. Noda,"Simple and secure password authentication protocol (SAS)," *IEICE Trans. Commun.*, vol.E83-B, no.6, pp.1363-1365, June 2000.
- [7] C. Lin, H. Sun, and T. Hwang,"Attacks and solutions on strong-password authentication," *IEICE Trans. Commun.*, vol.E84-B, no.9, pp.2622-2627, September 2001.
- [8] C. M. Chen and W. C Ku,"Stolen-verifier attack on two new strong-password authentication protocols," *IEICE Transactions on Communications*, vol.E85-B, no.11, pp.2519–2521,November 2002.

- [9] T. Tsuji, and A. Shimizu, "An impersonation attack on one-time password authentication protocol OSPA," *IEICE Technical Report*, ISEC2002-81, vol.102, no.436, pp.67-72, November 2002.
- [10] C. W. Lin, J. J. Shen, and M. S. Hwang, "Security enhancement for optimal strong password authentication protocol," *ACM Operating Systems Review*, vol.37, no.2, pp.7-12, April 2003.
- [11] W. C. Ku, H. C Tsai, and S. M Chen, "Two simple attacks on LSH's strong password authentication protocol," *ACM operating systems review*, vol.37, no.4, pp.26-31, 2003.
- [12] H. Y. Chien and J. K. Jan, "Robust and simple authentication protocol," *The Computer Journal*, vol.46, no.2, pp.193-201, 2003.
- [13] W. C. Ku, H. C. Tsai, and M. J. Tsaur, "A Common Weakness of Password Authentication Schemes Requires Synchronous Update", *Int. Computer Symposium*, Dec. 15-17, 2004, Taipei, Taiwan
- [14] W. C. Ku, " A hash based strong password authentication without using smart cards," *ACM Operating Systems Review*, vol.38, issue.1, pp.29-34, Jan. 2004.
- [15] T. H. Chen, W.B Lee, and G. Horng, "Secure SAS-like password authentication schemes," *Comput. Stand. Interfaces*, vol.27, no.1, pp.25-31, Nov. 2004
- [16] Y.F. Chang and C. C. Chang, "A secure and efficient strong-password authentication protocol," *ACM SIGOPS Operating Systems Review*, vol.38, no.3, pp.79-90, July 2004.
- [17] M. Kim and C. K. Koc, "A simple attack on a recently introduced hash

based strong password authentication scheme,” *International Journal of Network Security*, vol.1,no.2,pp.77-80, Sept. 2005.

- [18] H. Y. Chen, R. C. Wang, and C. C. Yang, “Note on Robust and Simple Authentication Protocol,” *The computer journal*, vol.48, no.1, 2005.
- [19] Y.F. Chang and C. C. Chang, “An Improvement on Strong-Password Authentication Protocols, ” *Embedded Software and Systems, Lecture Notes in Computer Science* vol. 3820, pp 629-637, 2005.
- [20] E. J. Yoon, E. K. Ryn, and Y. K. Young, “Fixing problems in Lin et al.’s OSPA protocol,” *Applied Mathematics and Computation* 166(1), pp.46-57, 2005.
- [21] H. C. Wu, M. S. Hwang, and C. H. Liu, “A secure and strong password authentication protocol,” *Journal Fundamental Informaticae*, vol.64, issue.4, pp.399-406, June 2005.
- [22] E. J. Yoon and K. Y. Yoo, “Robust Secret Key Based Authentication scheme using smart card,” *Advances in Multimedia Information Processing, Lecture Notes in Computer Science*, vol.3768, pp 723-734, 2005.
- [23] V. Goyal, V. Kumar, M. Singh, A. Abraham, and S. Sanyal, “A new protocol to counter online dictionary attacks,” *Computers & Security*, vol.25, issue 2, pp.114–120, March 2006.
- [24] C. W. Lin, C. S. Tsai, and M. S. Hwang, “A New Strong Password Authentication Scheme using One-Way Hash Functions,” *Journal of Computer and Systems Sciences International*, vol.45, issue.4, pp 623-626, July–August 2006.

- [25] C. C. Chang and H. C. Tsai, "A smart card based authentication protocol for strong passwords," *Proceedings of the 5th WSEAS International Conference on Applied Computer Science*, Hangzhou, China, pp385-391, April 2006.
- [26] K. Mangipudi and R. Katti, "A hash based strong password authentication protocol with user anonymity," *International Journal of Network Security*, vol.2, no.3, pp.205-209, May 2006.
- [27] C. S. Tsai, C. C. Lee, and M. S. Hwang, "Password authentication schemes: current status and key issues," *International Journal of Network Security*, vol.3,no.2,pp.101-115, Sept. 2006.
- [28] C. J. Mitchell and S. L. Ng, "Comments on the security of the SPAPA strong password authentication protocol," unpublished.
- [29] X. Tian, R. W. Zhu, and D. S. Wong, "Improved Efficient Remote User Authentication Schemes," *International Journal of Network Security*, vol.4, no.2, pp.149–154, March 2007.
- [30] N. S. Weragama and M. Sandirigama, "SAS-3: A polynomial based strong password authentication protocol," *ICIIS*, pp.41-46, Aug. 2007.
- [31] S. K. Sood, A. K. Sarje, and K. Singh, "Cryptanalysis of password schemes: current status and key issues," *ICM2CS*, pp.1-7, Dec. 2009.
- [32] Y. Lee and D. Wou, "Enhancing of a password based authentication scheme using smart cards," *On the Move to Meaningful Internet Systems, LNCS*, vol.5871, pp.879-886, 2009.
- [33] L. T. Liang and J. ZhiGang, "A new low cost One Time ID and Password Authentication Protocol using popular removable storage devices," *IEEE*

Conference on Intelligent Networks and Intelligent Systems, pp.213-216,
Nov. 2009.

- [34] M. Kumar, "On the security vulnerabilities of a hash based strong password authentication scheme," *Electronic print in International Association for Cryptologic Research*, 2010.
- [35] Y. Jingbo and S. Pingping, "A secure strong password authentication protocol," *IEEE Conference on Software Technology and Engineering*, vol.2, pp. 355-357, Oct. 2010.
- [36] J. Huiping, "Strong password authentication protocol," *IEEE Conference on Distance Learning and Education*, vol. 4, pp. 50-52, Oct. 2010.
- [37] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "The quest to replace passwords: a framework for comparative evaluation of web authentication schemes," *IEEE Symposium on Security and Privacy*, pp.553-567, May 2012.
- [38] C. C. Lee, C. H. Liu and M. S. Hwang, "Guessing Attack on Strong Password Authentication protocol," *International Journal of Network Security*, vol.15, no.1, pp.64-67, Jan. 2013
- [39] H. M. Elkamchouchi and M. H. Eldefrawy, "Weaknesses of Wu-Hwang-Liu's Password Authentication Protocol," *International Journal of Computer Science and Electronics Engineering*, vol.1, issue. 5, 2013
- [40] X. Zhuang, C. C. Chang, Z. H. Wang, and Y. Zhu, "A simple password authentication scheme based on geometric hashing function," *International Journal of Network Security*, vol.16,no.3,pp.237-243, May 2014.