Expansions of CHAP
Modificationless on Its Structures of Packet and Data Exchange

Masaki Inamura
Division of Information System Design, Department of Science and Engineering, Tokyo Denki University,
Ishizaka, Hatoyama-machi, Hiki-gun, Saitama, Japan

Keywords: Authentication, CHAP, Two-factor authentication, Administration free, Anonymous authentication.

Abstract: I propose new authentication protocols using/unmodifying the structures of packet and data exchange on CHAP (Challenge Handshake Authentication Protocol). CHAP is one of the most popular authentication protocols because of very simple scheme and no vulnerability of its structures of packet and data exchange. Therefore, this protocol is used a lot of services over the Internet as *de facto* standard. However, unlawful access often happens caused by users' mistakes of password setting, e.g. password-list-attacks, and proposing/implementation of other authentication methods without using password are urgent. To solve the problem, I propose new scheme which can send many type of authentication codes using intact CHAP. By using my proposal, other authentication method using password authentication together can be realized with a minimum cost burden.

1 INTRODUCTION

1.1 Weakness of Password Authentication

Recently, the Intranet becomes common technology and makes it possible for users to take many services over IP, e.g. multi party communication, shopping, banking, e-government service, and so on. In these services, a user authentication method is introduced because of protection against spoofing identity.

The most popular method of user authentication is *password authentication*. Especially, CHAP (Challenge Handshake Authentication Protocol) (Simpson, 1996) is generally used, e.g. authentication over PPP (Point-to-Point Protocol) (Simpson, 1994). However, the authentication used original CHAP is the following weakness:

- Users tend to use easy-to-guess password or the same password in multiple accounts. Therefore, if user authentication is dependent on only password, the system may be easy to be attacked.
- The administrator have to administer all users. Therefore, if an adversary intrudes the service system, users' privacy and/or personal information may be leaked.

Furthermore, in the recent ubiquitous computing, users can connect their computers/mobile terminals to the Internet anywhere. Therefore, they need to pay attention to the attacks from adversaries anytime (Sklavos and Zhang, 2007).

1.2 Related Work

Regarding the above weakness in section 1.1, some authentication methods/protocols except to use classical passwords have been proposed. One is *two-factor authentication* protocols (Schneier, 2005; Hagalisletto and Riiber, 2007; Aloul et al., 2009; Rathgeb and Uhl, 2010; Fan et al., 2010; Eldefrawy et al., 2011; Acharya et al., 2013; Hwang and Gope, 2014) for countermeasures against the former weakness, and the other is *anonymous authentication* protocols (Kilian and Petrank, 1998; Ateniese and Tsudik, 1999; Boneh and Franklin, 1999; Camenisch et al., 2006; Wachsmann et al., 2010; Au et al., 2013) for countermeasures against the latter weakness.

However, these protocols cannot use the structurs of the packet and data exchange of CHAP, and new other machines have to be established. Therefore, it entails many costs to introduce a new authentication protocol.

Inamura M..
Expansions of CHAP - Modificationless on Its Structures of Packet and Data Exchange.
DOI: 10.5220/0005228002130220
In *Proceedings of the 1st International Conference on Information Systems Security and Privacy* (ICISSP-2015), pages 213-220
ISBN: 978-989-758-081-9

1.3 Outline of Proposal

For solution of the above problem in section 1.2, I propose new authentication protocols.

My proposed protocols realize two types of authentication within the architecture of CHAP. One is two-factor authentication shown in section 3.1, which can authenticate users with two secret codes at the same time. Furthermore, this protocol can be customized to the existing system, whose server cannot be stopped and replaced, with introduction of only a proxy server as shown in section 3.2. And the other is anonymous authentication shown in section 4, which does not need to administer ID/password in the server.

These protocols modify only the calculation method of response with using keyed one-way hash function, e.g. HMAC (Hash-based Message Authentication Code) (Krawczyk et al., 1997), and symmetric key encryption algorithm, and do not need to customize/modify the sequence and packet format newly.

Therefore, network machines, e.g. access-point, router, gateway and so on, can be utilized without changing, and introduction costs of authentication protocol except passwords can be reduced.

2 CHALLENGE HANDSHAKE AUTHENTICATION PROTOCOL

2.1 Protocol

2.1.1 Symbols

I define symbols used in CHAP as follows:

ID: User identification code.

PW: Secret code for authentication (e.g. password).

C: Challenge code generated by random number generator.

R: Response code for the challenge *C*.

$H_K(\alpha,\beta)$: Keyed one-way hash function (hashing data β with a key α).

2.1.2 Preconditions

I describe preconditions for using CHAP as follows:

- The server, which authenticates users, is trusted party.
- *PW* sharing between the server and one user is not leaked out.

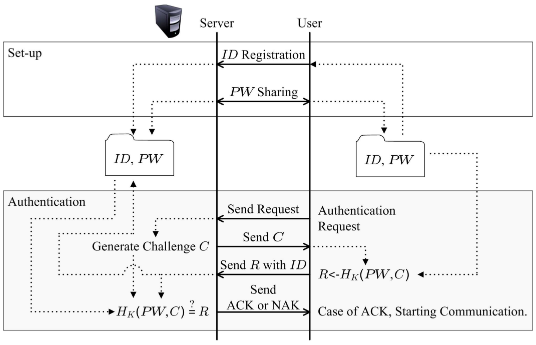


Figure 1: CHAP sequence.

- Third party can obtains only traffic packet and cannot know other code/data excepting the packet.

2.1.3 Procedure Sequence

CHAP has two phases ; one is “Set-up” and another is “Authentication.” I show these procedure sequence in figure 1 and describe the procedures as follows:

Set-up:

1. For registration, a user sends *ID* to the server.
2. The user generates *PW* and shares it with the server.
3. The server administer *ID* bound to *PW*.

Authentication:

1. For request, the user sends authentication request to the server.
2. The server generates *C* and sends it to the user.
3. The user calculates $R \leftarrow H_K(PW,C)$ and sends it with *ID* to the server.
4. The server calculates $H_K(PW,C)$ using *C* and administered *PW* bound to *ID*. The server verifies whether this generated value and received *R* from the user are equivalent or not, and inform the user about the result of this verification (“ACK” means the success of this verification, and “NAK” means the failure of this verification).

2.1.4 Packet Format

I show the packet format of CHAP in figure 2.

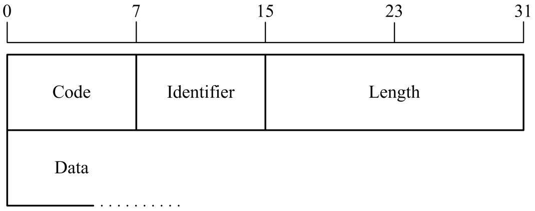


Figure 2: Packet format of CHAP.

The first octet (from 0th-bit to 7th-bit) means the code, which show the type of packet. There are four codes as follows:

0x01: Challenge.

0x02: Response.

0x03: Success.

0x04: Failure.

Next, the second octet (from 8th-bit to 15th-bit) means the identifier. This field aids in matching challenges, responses and replies.

The third and fourth octet (from 16th-bit to 31st-bit) means the length. This field and indicates the length of the CHAP packet including the code, identifier, length and data fields. Octets outside the range of the length field should be treated as data link layer padding and should be ignored on reception.

The octet after the fifth (from 32nd-bit) means the data. The format of this field is determined by the code field. However, in case of packet, which means success or failure, this field may be zero octets.

In four type of CHAP packet as above, the response is the most important packet as security, because security function is used in only this packet. I show the format of the response packet in figure 3.

The fifth octet (from 32nd-bit to 39th-bit) means the value length, which indicates the length of the value field.

The octets after the sixth (from 40th-bit) means the value, which has the fields indicated the value length field. The value in response packet is the one-way hash function calculated over a stream of octets consisting of the identifier followed by the secret code *PW* and the challenge code *C*. The length of the value depends upon the algorithm of the one-way hash function used in the authentication system.

The last octets after the value field means name. This field represents the identification of the system transmitting the packet. There are no limitations on the content of this field, e.g. ASCII character strings or globally unique identifiers in ASN.1 syntax[1]. However, the name should not be NULL or CR/LF terminated. The length of this field is determined from the length field and the value length field.

2.2 Security Analysis

2.2.1 Password Unleakability

When authentication, raw password *PW* is not send over the Internet. Information regarding *PW* is only send in the response packet. However, response value *R* is calculated from *PW* and the challenge code *C* with the one-way hash function. If security conditions (Rogaway and Shrimpton, 2004) of this one-way hash function are established, nobody can calculate *PW* from *R*.

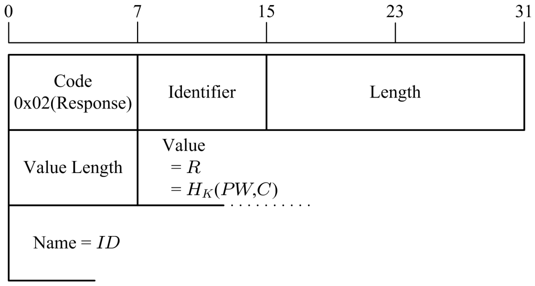


Figure 3: Response packet format of CHAP.

2.2.2 Unforgeability

When authentication, the challenge code *C* is generated from random generator. *C* is new value each time of authentication. Therefore, an adversary cannot reuse *C*, which has been sent over the Internet already, for forgery by replay attack.

3 TWO-FACTOR AUTHENTICATION OVER CHAP

3.1 Prototype Protocol

3.1.1 Symbols

I define symbols used in the prototype protocol as follows:

ID: User identification code.

$ID_{\mathbf{2nd}}$: Other identification code excepting *ID*.

PW: Secret code for authentication (e.g. password).

SK: Secret code for authentication excepting *PW* (e.g. encryption key).

C: Challenge code generated by random number generator.

R: Response code for the challenge *C*.

$H_K(\alpha,\beta)$: Keyed one-way hash function (hashing data β with a key α).

[1] These identifiers are defined as ASN.1 Project by *International Telecommunication Union Telecommunication Standardization Sector*.

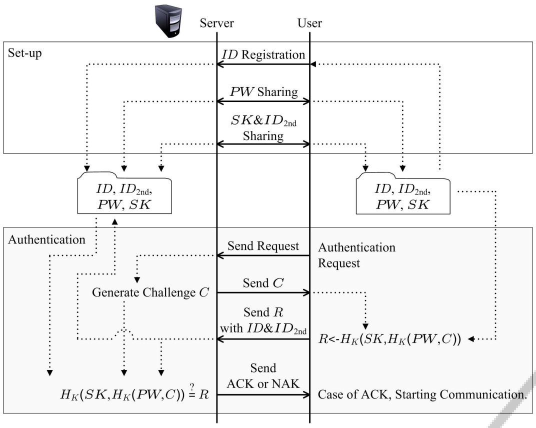


Figure 4: The sequence of prototype protocol of two-factor authentication over CHAP.

3.1.2 Preconditions

I describe preconditions for using the prototype protocol of two-factor authentication over CHAP as follows:

- The server, which authenticates users, is trusted party.
- *PW* and *SK* sharing between the server and one user is not leaked out.
- Third party can obtains only traffic packet and cannot know other code/data excepting the packet.

3.1.3 Procedure Sequence

I show "Set-up" and "Authentication" procedure sequence of the prototype protocol of two-factor authentication over CHAP in figure 4 and describe the procedures as follows:

Set-up:

1. For registration, a user sends *ID* to the server.
2. The user generates *PW* and shares it with the server.
3. The user generates ID_{2nd} and *SK* in addition to tha above and shares them with the server.
4. The server administer *ID* bound to *PW* and ID_{2nd} bound to *SK*.

Authentication:

1. For request, the user sends authentication request to the server.
2. The server generates *C* and sends it to the user.
3. The user calculates $R \leftarrow H_K(SK, H_K(PW, C))$ and sends it with *ID* and ID_{2nd} to the server.
4. The server calculates $H_K(SK, H_K(PW, C))$ using *C*, *PW* bound to *ID* and *SK* bound to ID_{2nd}. The server verifies whether this generated value and received *R* from the user are equivalent or not, and inform the user about the result of this verification ("ACK" means the success of this verification, and "NAK" means the failure of this verification).

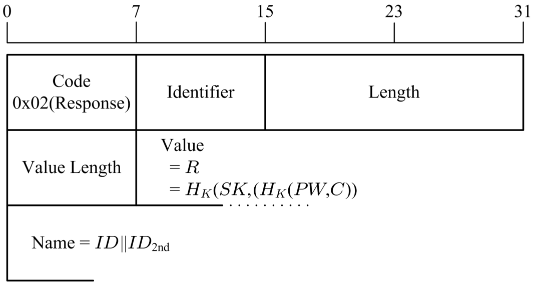


Figure 5: Response packet format regarding the prototype protocol of two-factor authentication over CHAP.

3.1.4 Response Packet Format

I show the format of the response packet regarding the prototype protocol of two-factor authentication over CHAP in figure 5.

The value length field is the same as that of the original response packet.

The octets after the sixth (from 40th-bit) means the value, which has the fields indicated the value length field. The value in response packet is the one-way hash function calculated over a stream of octets consisting of the identifier followed by the secret code *PW* and the challenge code *C*. The length of the value depends upon the algorithm of the one-way hash function used in the authentication system.

The last octets after the value field means name. This field represents the identification of the system transmitting the packet. There are no limitations on the content of this field, e.g. ASCII character strings or globally unique identifiers in ASN.1 syntax. However, the name should not be NULL or CR/LF terminated. The length of this field is determined from the length field and the value length field.

3.2 Revised Protocol with Proxy Server

3.2.1 Symbols

I define symbols used in the revised protocol as follows:

ID: User identification code.

ID_{2nd}: Other identification code excepting *ID*.

PW: Secret code for authentication (e.g. password).

SK: Secret code for authentication excepting *PW* (e.g. encryption key).

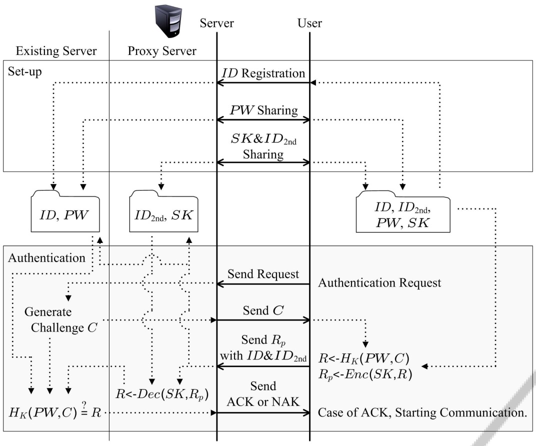


Figure 6: The sequence of revised protocol of two-factor authentication over CHAP.

C**:** Challenge code generated by random number generator.

R**:** Original response code for the challenge C.

R_p**:** Encryption code of the response conde R.

$H_K(\alpha,\beta)$**:** Keyed one-way hash function (hashing data β with a key α).

$Enc(\alpha,\beta)$**:** Encrypting function of symmetric key encryption Algorithm (Encrypting data β with a key α).

$Dec(\alpha,\beta)$**:** Decrypting function of symmetric key encryption Algorithm (Decrypting data β with a key α).

3.2.2 Preconditions

I describe preconditions for using the revised protocol of two-factor authentication over CHAP as follows:

- The existing server, which authenticates users, and the proxy server, which decrypts data. are trusted parties.
- *PW* sharing between the existing server and one user and *SK* sharing between the proxy server and one user are not leaked out.
- Third party can obtains only traffic packet and cannot know other code/data excepting the packet.

3.2.3 Procedure Sequence

I show "Set-up" and "Authentication" procedure sequence of the revised protocol of two-factor authentication over CHAP in figure 6 and describe the procedures as follows:

Set-up:

1. For registration, a user sends *ID* to the existing server.
2. The user generates *PW* and shares it with the existing server.
3. The user generates ID_{2nd} and *SK* in addition to tha above and shares them with the proxy server.
4. The existing server administer *ID* bound to *PW*, and The proxy server administer ID_{2nd} bound to *SK*.

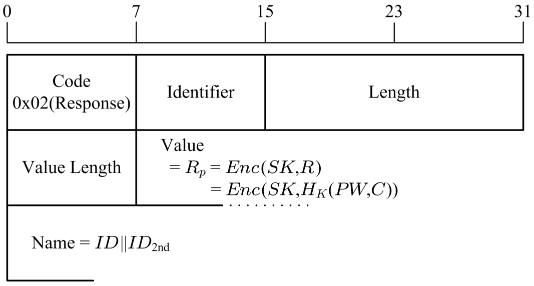
Authentication:

1. For request, the user sends authentication request to the existing server.
2. The existing server generates *C* and sends it to the user.
3. The user calculates $R \leftarrow H_K(PW,C)$. Furthemore, the user calculates $R_p \leftarrow Enc(SK,R)$. and sends it with *ID* and ID_{2nd} to the proxy server.
4. The proxy server calculates $R \leftarrow Dec(SK,R_p)$ using *SK* bound to ID_{2nd} and informs the existing server of it with *ID*.
5. The existing server calculates $H_K(PW,C)$ using *C* and *PW* bound to *ID*. The existing server verifies whether this generated value and received *R* from the user are equivalent or not, and inform the user about the result of this verification ("ACK" means the success of this verification, and "NAK" means the failure of this verification).

3.2.4 Response Packet Format

I show the format of the response packet regarding the revised protocol of two-factor authentication over CHAP in figure 7.

The value length field is the same as that of the original response packet.



Code 0x02(Response)

Identifier

Length

Value Length

Value $= R_p = Enc(SK,R)$ $= Enc(SK,H_K(PW,C))$

Name $= ID \| ID_{2nd}$

Figure 7: Response packet format regarding the revised protocol of two-factor authentication over CHAP.

3.3 Security Analysis

3.3.1 Secret Codes Unleakability

In prototype protocol, there are two secret codes: both *PW* and *SK* are the codes for password authentication. *PW* and *SK* are only used in the response packet. Therefore, an adversary can obtains $R \leftarrow H_K(SK, H_K(PW,C))$. However, $H_K(PW,C)$ can calculated from R because of features of general one-way hash function shown in section 2.2. Furthermore, Granted that an adversary can obtain $H_K(PW,C)$, *PW* cannot be calculated from $H_K(PW,C)$ similarly.

Also in revised protocol, there are two secret codes: *PW* is the code for password authentication, and *SK* is the key for symmetric key encryption. *PW* and *SK* are only used in the response packet. Therefore, an adversary can obtains many response packets, namely many types of cipher text $R_p \leftarrow Enc(SK,R)$. However, if security conditions (Bellare et al., 1997) of this symmetric key encryption are established, any adversaries cannot decrypt R_p or obtain R from the response value. Furthermore, Granted that an adversary can obtain R, *PW* cannot be calculated from R because of features of general one-way hash function shown in section 2.2.

3.3.2 Unforgeability

When authentication, the challenge code C is generated from random generator similar to original CHAP in section 2.1. Also this code is new value each time of authentication.

As a result, in prototype protocol, $H_K(PW,C)$ in the response packet is new value each time. Therefore, adversary cannot reuse C and $H_K(PW,C)$, which has been sent over the Internet already, for forgery by replay attack.

Furthermore, in revised protocol, also $H_K(PW,C)$ in the response packet is new value each time. Therefore, adversary cannot reuse C for password authentication and $H_K(PW,C)$ for the challenge code decryption, which has been sent over the Internet already, for forgery by replay attack.

3.4 Discussion

Regarding proposed protocols in section 3.1 and 3.2, on the one hand, there are the following strong points:

- If both of secret codes, which are *PW* and *SK*, are not legitimate, this authentication certainly fails. Therefore, the proposed protocols of the certification possess high reliability.
- Each protocol does not need to change its packet format. Therefore, hardware of communication, e.g. access-point, router, gateway and so on, and/or servers does not need to be replaced.
- Furthermore, regarding the proposed protocol in section 3.2, the existing system can be used just to put one proxy server. Therefore, two-factor authentication can be installed at low cost.

On the other hand, these proposed protocols have the following problems:

- If authentication fails, the administrator cannot know which secret code, *PW* or/and *SK*, is irregular.
- Even though the cost of installing hardware is low, it costs a few revised expenditure to introduce software.

4 AUTHENTICATION WITH ADMINISTRATION FREE OVER CHAP

4.1 Protocol

4.1.1 Symbols

I define symbols used in CHAP as follows:

*ID***:** User identification code.

*MK***:** Secret master key holding only server.

*PW***:** Secret code for authentication generated from *MK*.

PW'**:** Authentication code generated from *MK* in authentication phase.

*C***:** Challenge code generated by random number generator.

*R***:** Response code for the challenge *C*.

$H_K(\alpha,\beta)$**:** Keyed one-way hash function (hashing data β with a key α).

4.1.2 Preconditions

I describe preconditions for using the authentication with administration free over CHAP as follows:

- The server, which authenticates users, is trusted party.
- *PW* sharing between the server and one user is not leaked out.
- *MK* holding only the server is not leaked out.
- Third party can obtains only traffic packet and cannot know other code/data excepting the packet.

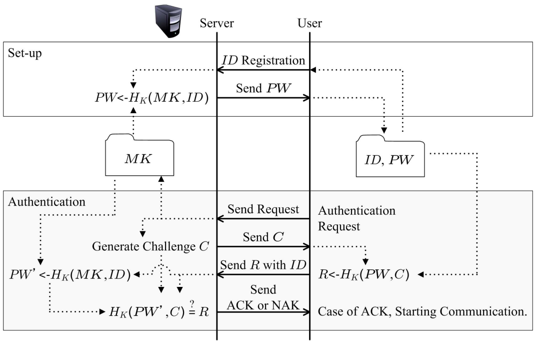


Figure 8: The sequence of authentication with administration free over CHAP.

4.1.3 Procedure Sequence

I show "Set-up" and "Authentication" procedure sequence of the authentication with administration free over CHAP in figure 8 and describe the procedures as follows:

Set-up:

1. For registration, a user sends ID to the server.
2. The server generates $PW \leftarrow H_K(MK, ID)$ and sends it to the user.
3. The user holds PW bound to PW.

Authentication:

1. For request, the user sends authentication request to the server.
2. The server generates C and sends it to the user.
3. The user calculates $R \leftarrow H_K(PW, C)$ and sends it with ID to the server.
4. The server calculates $PW' \leftarrow H_K(MK, ID)$ and $H_K(PW', C)$ using C. The server verifies whether this generated value and received R from the user are equivalent or not, and inform the user about the result of this verification ("ACK" means the success of this verification, and "NAK" means the failure of this verification).

4.1.4 Packet Format

The format of the response packet regarding the authentication with administration free over CHAP is the same as that of the original response packet.

Therefore, I omit the detailed description of this packet.

4.2 Security Analysis

4.2.1 Secret Master Key Unleakability

PW is calculated from ID and MK. ID can be generated by any user and opened in public. Therefore, if the user, who has already registered ID, can calculated MK from my PW, he/she can forge the authenticable pair of user identification code and secret code for authentication.

However, PW is calculated with the one-way hash function. If security conditions (Rogaway and Shrimpton, 2004) of this one-way hash function are established, any user cannot calculate MK from PW.

4.2.2 Secret Codes Unleakability

When authentication, information regarding PW is only send in the response packet similar to original CHAP in section 2.1.

Therefore, any adversaries cannot calculate PW from R because of the same reason in section 2.2.

4.2.3 Unforgeability

When authentication, the challenge code C is generated from random generator similar to original CHAP in section 2.1. C is new value each time of authentication. Therefore, an adversary cannot reuse C, which has been sent over the Internet already, for forgery by replay attack.

4.3 Discussion

Regarding proposed protocol in section 4.2, on the one hand, there are the following strong points:

- The administrator administers only his/her secret master key MK. He/she does not need to administer any users' identification code and secret code.
- Unless otherwise leaked the secret master key, the administrator can authenticate only legal users.
- Users' information is not stored in the server. Therefore, the anonymity of the user is kept when authentication, i.e. an anonymous authentication system over CHAP can be realized. Furthermore, even if adversaries attack the server, users' information of privacy is not leaked.
- This protocol does not need to change its packet format. Therefore, hardware of communication, e.g. access-point, router, gateway and so on, does not need to be replaced.

On the other hand, these proposed protocols have the following problems:

- Because all of users' secret codes PW is generated from the secret master key MK holding only the administrator and the identification code ID, all users cannot decide their own PW.

- Because the administrator does not administer user's identification code *ID*, he/she cannot distinguish a user requesting authentication from other users.
- If user's secret code *PW* is leaked and needs to be reissued, his/her identification code *ID* have to be renewed.
- If the master key *MK* is leaked and needs to be reissued, all of users' secret codes *PW* have to be renewed

5 CONCLUSION

In this paper, I have proposed two types of authentication protocol revised CHAP; one is the two-factor authentication, and another is the authentication with administration free. Both use original format and sequence of CHAP and do not need to substantial revision to existing system. Therefore, new authentication protocols can be installed securely and easily with few costs.

As a future work, I plan to make the simulation systems installing these proposed protocols and measure these performances. Furthermore, I plan to propose other authentication protocols over CHAP.

REFERENCES

Acharya, S., Polawar, A., and Pawar, P. Y. (2013). Two factor authentication using smartphone generated one time password. *IOSR J. Computer Engineering*, 11(2):85–90.

Aloul, F. A., Zahidi, S., and El-Hajj, W. (2009). Two factor authentication using mobile phones. In *IEEE/ACS International Conference on Computer Systems and Applications - AICCSA 2009*, pages 641–644. IEEE press.

Ateniese, G. and Tsudik, G. (1999). Some open issues and new directions in group signatures. In *International Conference on Financial Cryptography - FC '99*, volume LNCS 1648, pages 196–211. Springer.

Au, M. H., Susilo, W., Mu, Y., and Chow, S. S. M. (2013). Constant-size dynamic *k*-times anonymous authentication. *IEEE Systems J.*, 7(2):249–261.

Bellare, M., Desai, A., Jokipii, E., and Rogaway, P. (1997). A concrete security treatment of symmetric encryption. In *Annual Symposium on Foundations of Computer Science - FOCS '97*, pages 394–403. IEEE Press.

Boneh, D. and Franklin, M. K. (1999). Anonymous authentication with subset queries (extended abstract). In *ACM Conference on Computer and Communications Security - CCS '99*, pages 113–119. ACM.

Camenisch, J., Hohenberger, S., Kohlweiss, M., Lysyanskaya, A., and Meyerovich, M. (2006). How to win the clone wars: Efficient periodic n-times anonymous authentication. *IACR Cryptology ePrint Archive*, Report 2006/454.

Eldefrawy, M. H., Alghathbar, K., and Khan, M. K. (2011). Otp-based two-factor authentication using mobile phones. In *International Conference on Information Technology: New Generations - ITNG 2011*, pages 327–331. IEEE press.

Fan, C., Ho, P., and Hsu, R. (2010). Provably secure nested one-time secret mechanisms for fast mutual authentication and key exchange in mobile communications. *IEEE/ACM Trans. Networking*, 18(3):996–1009.

Hagalisletto, A. M. and Riiber, A. (2007). Using the mobile phone in two-factor authentication. In *International Workshop on Security for Spontaneous Interaction - IWSSI 2007*.

Hwang, T. and Gope, P. (2014). Provably secure mutual authentication and key exchange scheme for expeditious mobile communication through synchronously one-time secrets. *Wireless Personal Communications*, 77(1):197–224.

Kilian, J. and Petrank, E. (1998). Identity escrow. In *Advances in Cryptology - CRYPTO '98*, volume LNCS 1462, pages 169–185. Springer.

Krawczyk, H., Bellare, M., and Canetti, R. (1997). Hmac: Keyed-hashing for message authentication. Request for Comments, RFC 2104.

Rathgeb, C. and Uhl, A. (2010). Two-factor authentication or how to potentially counterfeit experimental results in biometric systems. In *International Conference on Image Analysis and Recognition - ICIAR 2010*, volume LNCS 6112, pages 296–305. Springer.

Rogaway, P. and Shrimpton, T. (2004). Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In *International Workshop on Fast Software Encryption - FSE 2004*, volume LNCS 3017, pages 371–388. Springer.

Schneier, B. (2005). Two-factor authentication: Too little, too late. *Communications of the ACM*, 48(4):136.

Simpson, W. A. (1994). The point-to-point protocol (ppp). Request for Comments, RFC 1661.

Simpson, W. A. (1996). Ppp challenge handshake authentication protocol (chap). Request for Comments, RFC 1994.

Sklavos, N. and Zhang, X. (2007). *Wireless Security and Cryptography: Specifications and Implementations*. CRC-Press.

Wachsmann, C., Chen, L., Dietrich, K., Löhr, H., Sadeghi, A., and Winter, J. (2010). Lightweight anonymous authentication with tls and daa for embedded mobile devices. In *International Conference on Information Security - ISC 2010*, volume LNCS 6531, pages 84–98. Springer.