Two Lightweight Authenticated Key Agreement Protocols Using Physically Unclonable Function with Privacy Protection

Dan Zhu[1], Liwei Wang[2], and Hongfeng Zhu[2*]
(Corresponding author: Hongfeng Zhu)

School of Foreign Languages, Shenyang Jianzhu University[1]
No.9, HunNan East Street, HunNan District, Shenyang 110168, China
(zhudan413@163.com)
Software College, Shenyang Normal University[2]
No.253, HuangHe Bei Street, HuangGu District, Shenyang 110034, China
(1696751943@qq.com; zhuhongfeng1978@163.com)
(Received July 31, 2019; Revised and Accepted Dec. 3, 2019; First Online Apr. 6, 2020)

Abstract

With the continuous development of the information age, people's demand for information security is also getting higher and higher. In recent years, in order to authenticate unsafe communication more effectively, some people combine password with input Physically unclonable function (PUF). The PUF described in this authentication method has hardware-based embedding function and has important physical inconsistency in authentication. In this paper, we will continue to propose more effective authentication protocols based on PUF algorithm. In published papers, authentication involves only one user and one server. Today, we discuss a text authentication protocol involving three parties, which is a three-party key agreement protocol based on PUF algorithm. This key agreement protocol has more practical functions. The two users in the protocol and the server have different public keys and their private keys. After a series of calculations and the server's message transmission, the information in the hands of the two users is matched again. Based on continuous experiments, we find that this key agreement protocol is effective.

Keywords: Key Exchange; Mutual Authentication; Physically Unclonable Function; Privacy Protection

1 Introduction

In today's society, many people already love browsing on various websites. At the same time, with the development of the network, information security has been involved in people's production and life. At present, the most popular technology is authentication protocol based on some algorithm. Two password-based PUF authentications have been introduced in literature [1, 6]. First, a physical unclonable function (PUF) is physically embedded in a hardware of device and always outputs an unpredictable noise y for an input x depending on unique hardware characteristics. The PUF [5] also has an unclonable property that any attempt to clone or reproduce makes itself unrecoverable from an original one. Owing to its unpredictability and unclonability, in recent years, the PUF has been extensively integrated into devices over wireless communication environment.

From the most practical point of view, human memory password is one of the most common authentication methods in wireless personal communication, because this kind of password is the most convenient and simple authentication tool in practice. But in many cases, passwords in user memory are easily guessed or stolen, and they are inherently vulnerable to well-known online and online dictionary attacks [9]. Because this simple and simple memory password has been attacked and stolen, two factor authentication (smart card and password) have been designed [6]. Recently, due to the emergence of new hardware technologies such as PUF, some scholars have studied the combination of password and PUF [1] in order to conduct more effective authentication [7] in unsafe communications, and hope to prevent the loss of password and personal information in this way. Protecting personal privacy [3]. They are all based on a same assumption that the PUF is initially integrated with a fuzzy extractor (FE) for converting a PUF's unpredictable output into a stable output. It first takes a password input password from a user and outputs an unpredictable secret s through FE. The secret s later serves as a main authentication factor.

In this way, network adversaries can prevent guessing attacks on user's personal passwords. At present, the published literature on key agreement by combining password

and PUF involves only a single user and server Two existing schemes are based on an IUF, for example: PUF + PAKE Scheme, PUF + ZKPK Scheme [6].

The key agreement protocol based on PUF, which has been written into the literature above, has been solved. However, we still need to think about the next issues.

This paper attempts to design a new protocol, which can be established in a more practical environment under the existing PUF algorithm [4]. This protocol breaks the tradition and is no longer a single key agreement between users and servers. It is upgraded to a three-party key agreement protocol based on PUF algorithm [11, 12]. To this end, we will review the key authentication process based on PUF algorithm published in the literature. We will refer to the published literature [5–7]. On the basis of the elaboration, this paper discusses the definition, steps, practical uses, advantages and disadvantages of the three-party key agreement protocol based on PUF, and improvements.

Generally speaking, the purpose of this paper is as follows:

1) We design two key agreement protocols based on PUF algorithm, one is tripartite key agreement protocol, the other is group key agreement protocol. This scheme can support mutual authentication and ensure the security of authentication.

2) The two protocols designed by us can resist off-line password guessing attacks, and have strong practicality and security.

3) The two key agreement protocols designed by us can also protect perspicacious.

The arrangement of this paper is as follows: We will outline the preparatory knowledge in the second section. In the third section, three-party and group key agreement protocols based on PUF algorithm are introduced. Section 4 gives the analysis of security and efficiency. Section 5 gives a summary of this paper.

2 Related Work

Nowadays, there are two types of authentication protocols for PUF algorithm in published literature [1, 2]. One is PUF+PAKE scheme and the other is PUF+ZKPK scheme. The basic idea of these two protocols is to convert unpredictable PUF output into a unified random key that can be used as a key directly on the integrated PUF by using a fuzzy processor [6]. The most important thing is that these two authentication protocols are under the integrated PUF algorithm, and after dealing with noise through the fuzzy processor, they can always get a clearer output on the same input.

2.1 PUF and Password Combination

Firstly, we need to review the process of using PUF and password to generate keys for registration published in the literature. They all execute the registration protocol in the secure channel and then run the authentication protocol. The registration protocol of PUF+PAKE scheme [6] is showed in Figure 1.

The Server S randomly selects a random number c_i and sends it to the user U. Next, the user U calculates $d_i \leftarrow H(c_i \| pwd)$ with his own memory password.User U use input to calculate PUF and get (s_i, w_i) from the formula $Gen(PUF(d_i))$. The registration protocol of PUF+ZKPK scheme [6] is showed in Figure 2.

The Server S randomly selection of a random Value c, and chooses $\{G_q\}$. And send them to users U. U evaluates PUF for an input $H(H(c||pwd, \{G_q\})$ and calculates s, w from $Gen(PUF(H(H(c||ped), \{G_q\}))$ User U computes $u(= g^s mod q)$ and sends (u, w) to Server S, its corresponding list $(c, \{G_q\}, u, w)$ is maintained in DB.

2.2 Framework of Our Scheme

In Figure 3, we further illustrate the steps of our tripartite key agreement protocol. In this protocol, we define two user names: U_1 and U_2, and define the server as S. U_1 and S have an common key k_1, U_2 and S have an common key k_2. U_1 and U_2 choose their temporary private key, $x_1, y_1 = g^{x_1}$,$x_2, y_2 = g^{x_2}$. Then, U_1 computes $E_{k_1}(U_1||y_1)$, $E_{k_1}(U_1||U_2||y_2)$ and U_2 computes $E_{k_2}(U_2||y_2)$, $E_{k_2}(U_2||U_1||y_1)$. U_1 and U_2 uses S to distribute and transmit messages. At last, U_1 and U_2 computes $SK_{u_1u_2} = y_2^{x_1}$, $SK_{u_1u_2} = y_1^{x_2}$. The two formulas are equal or not.

3 The Improved Two-Party PAKA Protocol with Privacy Protection

3.1 Notations

The concrete notations used hereafter are shown in Table 1.

3.2 Authenticated Key Agreement Phase

Figure 4 illustrates the user Authenticated phase. When two users, one The server or three parties conduct key agreement, two users hold the shared key with the server respectively k_1, k_2.These two users and The servers will complete the authentication process on the secure information channel.

Step 1: User U_1, U_2 and The Server Pass Shared Key k_1, k_2 , U_1 randomly selection x_1 to compute private key $y_1 = g^{x_1}$, U_2 randomly selection x_2 to compute private key $y_2 = g^{x_2}$.

Step 2: After receiving the message k_1, k_2 from U_1, U_2, Two users using $E_{k_m}()$ encryption method to compute $E_{k_1}(U_1||y_1)$,$E_{k_2}(U_2||y_2)$. And pass the calculation result to the server. The server will trans-

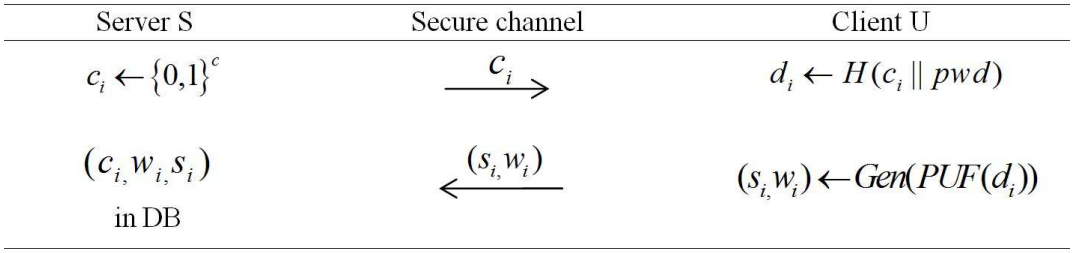

Figure 1: Enrollment phase in PUF + PAKE scheme

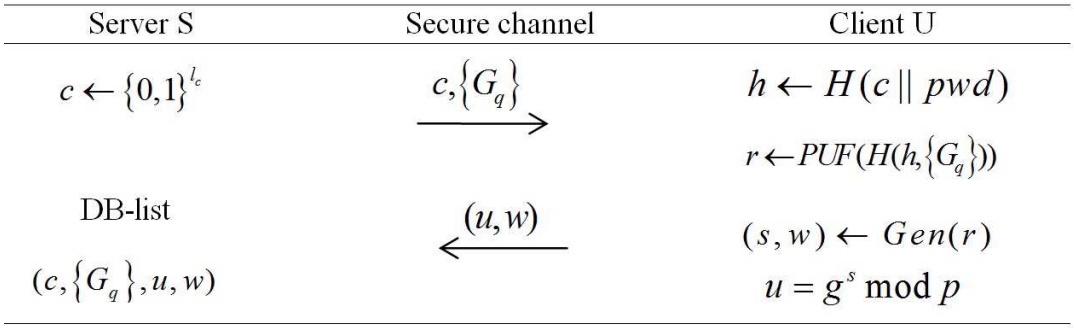

Figure 2: Enrollment phase in PUF + ZKPK scheme

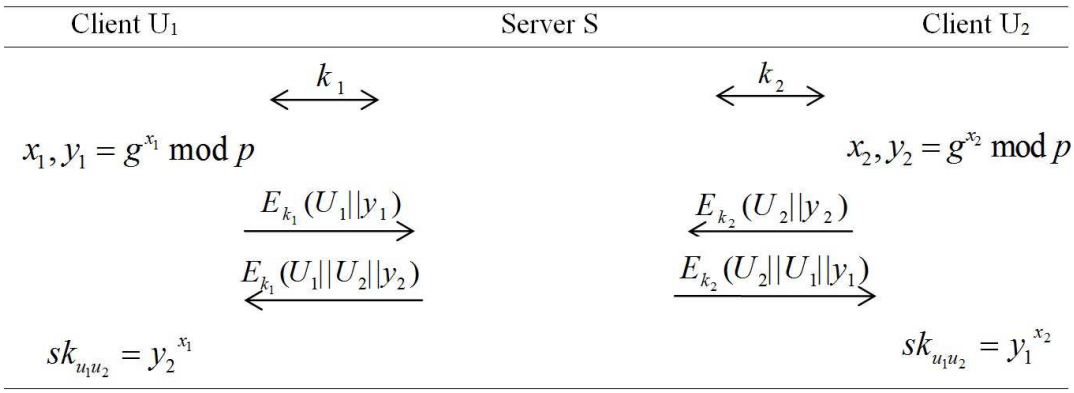

Figure 3: Authenticated key exchange phase of tripartite key agreement protocol

Table 1: Notations

Symbol	Definition
U_1, U_2,U_n	User name
S	Server
K_1,K_2,K_n	Shared Key between User and Server
x_1,x_2,y_1,y_2	private key
$E_{K_m}()$	Use $K_m$to symmetrically encrypt, m is a nonzero integer
$SK_{U_1U_2}$	Session key between U_1 and U_2
SK_{Group}	Group session key
$\|\|$	concatenation operation
$\oplus$	exclusive or operation
$H()$	Hash Functions

mit the results $E_{k_1}(U_1||U_2||y_2)$ and $E_{k_2}(U_2||U_1||y_1)$ to two users.

Step 3: In the above two steps, the server acts as a messaging role, Users decrypt and verify hash functions. They calculate the two results separately with the information they receive. $sK_{u_1u_2} = y_2^{x_1}$, and $sK_{u_1u_2} = y_1^{x_2}$. Then the two results are equal, the authentication is completed. If any authenticated process does not pass, the protocol will be terminated.

3.3 Authentication Phase of Multiparty Key Agreement Protocol

Based on the tripartite key agreement protocol, we continue to propose a group key agreement protocol scheme. The registration process of group key agreement protocol scheme is similar to that of tripartite key agreement protocol, which is based on PUF+ZKPK scheme. Figure 2 illustrates the user registration phase. The flow of group key agreement is illustrated in Figure 5 below.

Step 1: The Server S and User U_i extract K_i using PUF algorithms. Meanwhile, The Server S shares the key K_i with U_1,U_2,U_3 ……,U_n.

Step 2: The server S use shared keys with each user to compute, $T_i = K_1 \oplus K_2 \oplus K_3 \cdots \oplus K_{i-1} \oplus K_{i+1} + Timestamp$,and $MAC = H(U_1||U_2|| \ldots ||U_n||K_1 \oplus K_2 \oplus K_3 \cdots \oplus K_n||Timestamp)$. Then, the server broadcasts MAC and time stamp(K_i keep the same number with Time stamp) to the user remove U_i. And send T_i and MAC to U_i separately.

Step 3: U_i needs to compute $SK_i = K_i \oplus T_i$ (K_i keep the same number with T_i). The Server needs compute MAC_i locally. So next U compare MAC_i and MAC. If the two results are equal, the group session key is that $SK_{Group} = H(SK_i||Timestamp)$.

4 Security and Efficiency Analysis

In this section, we will describe a security model of three-party key agreement scheme and group key agreement scheme based on PUF algorithm. And we will prove that the computation of these two schemes is secure in random oracle and ideal cryptography models[8-10].

4.1 Provable Security of Tripartite Key Agreement Protocol

We use the following security model [8] to define the security requirements of authentication schemes for tripartite and group key agreement protocols.

Players. We define a server S and a user U who can participate in the authentication scheme certification of the key agreement protocol. Each of them has different instances. We call them oracles.

Queries. Adversary A can interact with participants and try to break Key or authentication of a user or server. For this purpose, we can use multiple queries.

1) Execute(U,S): This query simulates a passive attack in which the Adversary A will eavesdrop on the authentication communication process between the user U and the server S.

2) Reveal(I): This query simulates the abuse of session keys between instances I. Queries are available only if the attacked the instance I actually holds the session key and releases it.

3) Send(I,m): This query adversary A models the message sent to the instance I. Adversary A receives the response generated when the message m is processed according to the agreement p. In our scheme, adversary emphA query sending (S, start) initializes the key exchange algorithm, so adversary A receiving server should send the stream to the client.

4.2 Security Proof

The following theorem shows that the proposed scheme can safely distribute session keys under the assumption that it is reasonable and well-defined and more difficult to handle [9].

Theorem 1. *Let P be the above agreement and password be a finite dictionary of size N equipped with a uniform distribution. Let A be an adversary against the AKE security of P within a time bound t, with less than q_s interactions with the parties and q_p passive eavesdropping, and asking q_h hash-queries and qe encryption/decryption-queries. Then, we have* $\mathrm{Adv}_p^{ake}(A) \leqslant \frac{q_s}{N} + 4q_h \mathrm{Succ}_G^{cdh}(t') + \frac{(q_s+q_p)^2}{q-1} + \frac{(q_h+4q_s+q_p+q_e)^2}{2^\ell}$ *where* $t' \leqslant t + (q_s + q_p + 1) \cdot \Gamma_G$ *withℓ denoting $\ell_1, \ell_2, \ell_3, \ell_4$ and Γ_G denoting the computational time for an exponentiation in* G.

Proof. **Stage:** In this proof, for simplicity, we do not consider forward secrecy. We incrementally define sequence of games starting at the real game G_0 and G_1, G_2. For each G_n (n = 0, 1, 2) we define the following events:

1) S_n occurs if A correctly guesses the bit b involved in the *Test-query*.

2) $Encrypt_n$ occurs if A submits data it has encrypted by itself using the password.

3) $Auth_n$ occurs if A submits an authenticator $Auth$ that is accepted by the server and that has been built by the adversary itself.

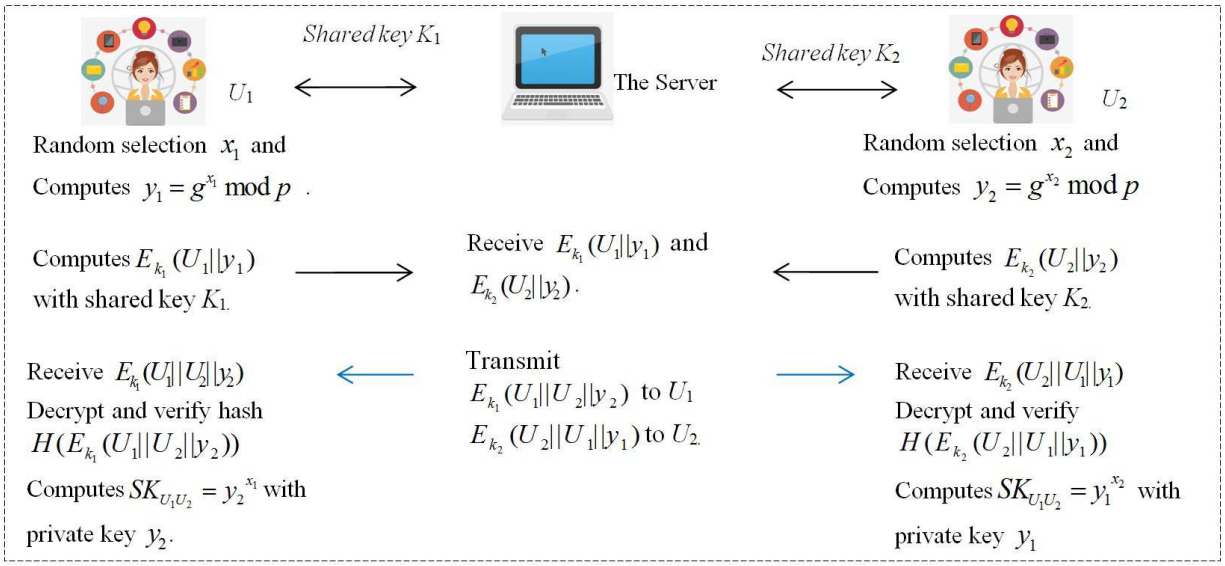

Figure 4: Authenticated key agreement phase

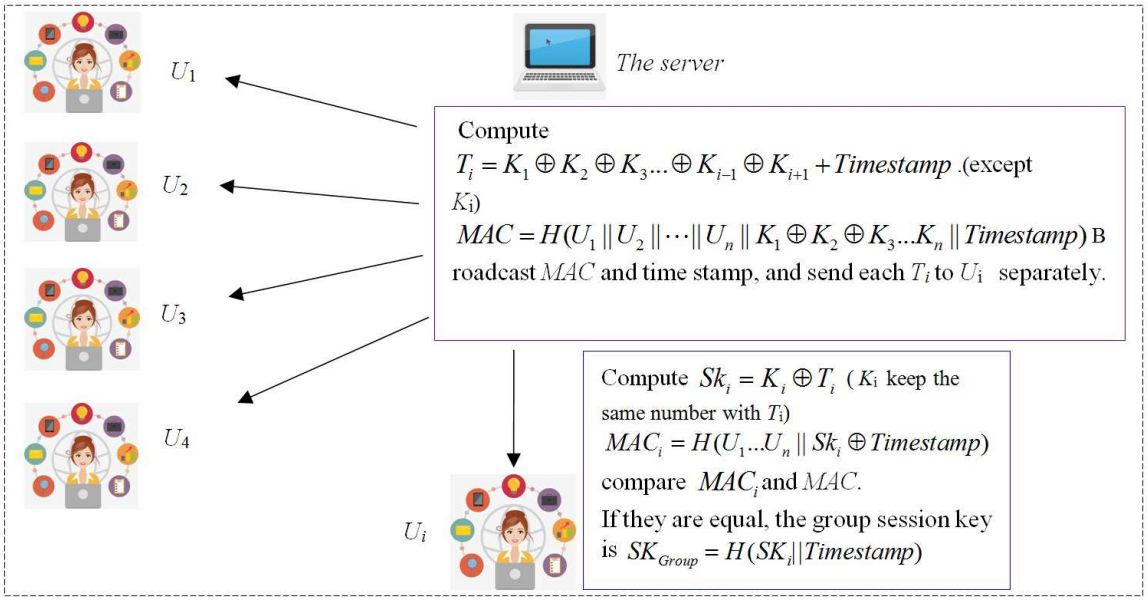

Figure 5: Authentication phase of multiparty key agreement protocol

Game. G_0 : This is the real agreement in the random oracle and ideal-cipher models. Several oracles are thus available to the adversary A: one hash oracles ($H()$) and all the instances U and S (in order to cover concurrent executions). We have

$$\mathrm{Adv}_p^{ake}(A) = 2P_r[S_0] - 1.$$

In the game below, we further assume that when the game aborts or stops with no answer b' outputted by adversary A we choose this bit b' at random, which in turn defines the actual value of the event S_K, Moreover, if the adversary A has not finished playing the game after q_s Send-queries or lasts for more than time t, we stop the game (and choose a random bit b'), where q_s and t are predetermined upper bounds.

Game. G_1 we simulate the hash oracles, $H()$, and five additional hash functions $H()$ and the encryption/decryption oracles and an encryption list.We also simulate all the instances, as the real players would, for the Send-queries and for the *Execute*, *Reveal* and *Test-queries*. From this simulation, it is clear that the game is perfectly. Thus, we have

$$P_r[S_1] = P_r[S_0].$$

Game. G_2 :In this game, the opponent guesses the session key without asking the corresponding Oracle h, so that it exists separately from the password and the temporary key, which are protected by the PUF algorithm. Otherwise, we will use the early game change method. Thus, we have

$$|P_r[S_1] - P_r[S_2]| \leqslant \frac{q_h^2}{2^{\ell+1}} + \frac{q_\varepsilon}{2^{\ell_4+1}} + \frac{(q_s+q_p)^2}{2(q-1)}$$

□

4.3 Further Security Discussion

1) The scheme could resist password guessing attack.

Proof. This attack means that adversary A will try to guess the password of the legitimate user based on the transmitted information. Password guessing attacks can only crack functions with a low-entropy variable (password), so we need to insert at least one large random variable that can withstand such attacks. In our protocol, when there is no transmission information, adversary A can only launch an online password guessing attack, using the password as the input value. Even if the opponent gets secret information, he does not have any comparative data to verify whether the password guess is correct without the help of the server. In other words, an adversary will not be able to construct tables. On the other hand, the maximum number of permissible invalid attempts for online password guessing attacks is only a few, and the account will be locked by the registered server [9, 10]. □

2) The scheme could support mutual authentication.

Proof. The Registration Server S verifies the authenticity of user U's request through validating the condition $MAC_i = MAC$ during the proposed phase. To compute $MAC_i = H(U_1 \ldots U_n || SK_i \oplus Timestamp)$, the attacker must has the password. Furthermore, MAC_i includes a large random nubmer *Timestamp*, the adversary cannot replay the old messages in the protocol. So, mutual authentication can successfully achieve in our scheme. □

3) The scheme could resist replay attack.

Proof. Validation messages include temporary random numbers, such as timestamps. More importantly, all such temporary random numbers are protected by corresponding information. Only legitimate users with secret keys and passwords can find these problems. □

4) The user-privacy protection can be provided in the proposed scheme.

Proof. There is no clear text in the authentication message sent in our proposed protocol. But authentication messages include overwritten ciphertext, which can send any important information to the other party or to a designated place using the public key of the peer, such as the identity in the scheme. Another message is to verify ciphertext using a one-way secure hash function. The other message is transmitted dynamically through channels and cannot be cloned. In addition, there is no duplicate message section in continuous communication. This shows that our scheme implements the attributes of user privacy. □

4.4 Efficiency Analysis

Table 2 shows some basic calculation processes of the three-party key agreement protocol and the group key agreement protocol based on the PUF algorithm written in this paper. In addition, compared with the two-key agreement protocol based on the PUF algorithm, this new protocol is more powerful and more computationally intensive. It is safer and more reliable in the safe transmission of information. Among them, $Ours_1$: three-party key agreement protocol based on PUF algorithm, $Ours_2$: PUF algorithm based group key agreement protocol. Yes/No: Support/Not support , T_{hash}: Time for executing the hash function.

5 Conclusion

This paper presents a tripartite and group key agreement protocol based on PUF algorithm. It extends on the basis of two-party key agreement protocol. This protocol not only inherits the advantages of the two-party key

Table 2: Comparisons between our proposed scheme and the related literatures

	PUF+PAKE [7]	PUF+ZKPK [7]	$Ours_1$	$Ours_2$
Shared secret key	No	No	Yes	Yes
Communication round	3round	2round	2round	1Broadcast
Vulnerable to replay attack	No	No	Yes	Yes
Formal security proof	No	No	Yes	Yes
Privacy protection	No	No	Yes	Yes
Authentication	Mutual	Mutual	Mutual	Mutual

agreement protocol, but also innovates. In addition to being able to authenticate between users and servers, the two-party key agreement protocol can also resist guessing attacks. This new protocol has the characteristics of resisting off-line password attack, supporting mutual authentication and providing privacy protection for users. The protocol also has strong security and enforce-ability, and enriches the types of key agreement protocols.

Acknowledgments

This work was supported by the Liaoning Provincial Natural Science Foundation of China (Grant No. 2019-MS-286), and Shenyang Science & Technology Innovation Talents Program for Young and Middle-aged Scientists (2019).

References

[1] D. Amelino, M. Barbareschi, E. Battista, A. Mazzeo, "How to manage keys and reconfiguration in WSNs exploiting sram based PUFs," in *Intelligent Interactive Multimedia Systems and Services*, pp. 109-119, 2016.

[2] F. Afghah, B. Cambou, M. Abedini, S. Zeadally, "A reram physically unclonable function (ReRAM PUF)-based approach to enhance authentication security in software defined wireless networks," *International Journal of Wireless Information Networks*, 2018. DOI: 10.1007/s10776-018-0391-6.

[3] M. Barbareschi, "Notions on silicon physically unclonable functions," in *Hardware Security and Trust*, pp. 189-209, 2017.

[4] J. W. Byun, I. R. Jeong, "Comments on physically unclonable function based two-factor authentication protocols," *Wireless Personal Communications*, vol. 106, no. 3, pp. 1243-1252, 2019.

[5] N. Chikouche, P. L. Cayrel, E. M. Mboup, *et al.*, J. Supercomput, "A privacy-preserving code-based authentication protocol for internet of things," *The Journal of Supercomputing*, vol. 75, no. 12, pp. 8231-8261, 2019.

[6] J. Delvaux, D. Gu, R. Peeters, and I. Verbauwhede, "A survey on lightweight entity authentication with strong PUFs," *ACM Computing Surveys (CSUR'15)*, vol. 48, no. 2, 2015.

[7] B. Halak, "Physically unclonable functions," *Electronics & Electrical Engineering*, 2018. ISBN 978-3-319-76804-5.

[8] P. Mall, M. Z. A. Bhuiyan, R. Amin, "A lightweight secure communication protocol for IoT devices using physically unclonable function," in *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, vol. 11611, pp. 26-35, 2019.

[9] D. I. Moon, A. Rukhin, R. P. Gandhiraman, B. Kim, M. Meyyappan, "Physically unclonable function by an all-printed carbon nanotube network," *ACS Applied Electronic Materials*, 2019. (`https://doi.org/10.1021/acsaelm.9b00166`)

[10] A. C. D. Resende, K. Mochetti, D. F. Aranha, "PUF-based mutual multifactor entity and transaction authentication for secure banking," in *Lightweight Cryptography for Security and Privacy*, vol. 9542, pp. 77 -96, 2015.

[11] D. P. Sahoo, "Design and analysis of secure physically unclonable function compositions," *Design and Analysis of Secure Physically Unclonable Function Compositions*, 2017. (`http://www.idr.iitkgp.ac.in/xmlui/handle/123456789/8243`)

[12] D. P. Sahoo, A. Bag, S. Patranabis, D. Mukhopadhyay, R. S. Chakraborty, "Fault-tolerant implementations of physically unclonable functions on FPGA," in *Security and Fault Tolerance in Internet of Things*, pp. 129-153, 2019.

Biography

Dan Zhu obtained her master degree in English Language and Literature from Liaoning Nornal University. Dan Zhu is a teacher at Shenyang Jianzhu University. She has research interests in Big Data, e-learning, and translation. Mrs. Zhu has published more than 10 international journal papers on the above research field.

Liwei Wang a postgraduate studying at Shenyang Normal University. She has researched interests in network security and quantum cryptography. Under the guidance of the teacher, she has published one article in

EI journals.

Hongfeng Zhu obtained his Ph.D. degree in Information Science and Engineering from Northeastern University. Hongfeng Zhu is a full professor of the Kexin software college at Shenyang Normal University. He is also a master's supervisor. He has research interests in wireless networks, social networks, network security and quantum cryptography. Dr. Zhu had published more than 60 international journal and international conference papers on the above research fields.