A Password-based Authenticator: P-Auth

Wei Linna
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, China
pna705@yahoo.com.cn

Qin Zhiguang
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, China
qinzg@uestc.edu.cn

***Abstract*—At present most of the existing authenticators require the input of high entropy keys. However, these keys are hard to remember and a great deal of the existing authentication protocols are based on passwords, the low entropy ones, which are given by users in practice. In this paper, we propose a password-based authenticator P-Auth. P-Auth uses "reputation" to generate high entropy unite secrets that are build on password and session key. It also employs threshold scheme to resist off-line dictionary attack and provide forward security.**

***Keywords*—authenticator, password, threshold scheme**

I. INTRODUCTION

In order to grab benefits an adversary in the communication channel has the willing and power to arbitrarily modify, delete, block or inject message between communicators. Thus, authentication protocols aiming at guarantee message source and avoid improper modification on the flows have been put forward in the interest of solving this problem. Now it also becomes one of the crucial secure communication issues [1， 2]. Numerous authentication protocols have been provided to against malicious adversaries. Nevertheless, due to lacking of sufficient specification and analysis flaws are often found in these designs. In [3] Bellare et al. proposed the notion of authenticator to simplify problems in authentication protocol design. In the paper they formalized two models. Authenticated-link model (AM), in which an adversary is limited to faithfully transmit a message from its sender to the receiver. That is, an adversary could do nothing beyond changing order of messages or simply drop them in this model. The unauthenticated-link model (UM), whereas, does not have such requirement. Given these two models, an authenticator is used as a "protocol compiler" to transform a proven secure protocol under AM into an equilibrium one in UM. By this way the intricate work of authentication protocol design and analysis is significantly reduced. Following their method, lots of message transmission authenticators (MT-authenticator) are proposed [4, 5, 6, 7]. Jiang et al. in [8] suggested an authenticator E-Auth that successfully transforms a protocol as a whole. Compared with those MT-authenticators which compile each message in protocol, his authenticator increases the round complexity of the resulting protocol only by at most an additively small number. However, Jiang's work requires some encryption scheme E with a public/private key pair (e_i, d_i) and it is well known that the issue of asymmetric key pair generation, distribution and maintenance is so complex that it itself is another branch in security realm [9]. As we know asymmetric key system is seldom used in people's daily life, we propose a password-based authenticator P-Auth in this paper.

P-Auth is an authenticator based on MT-authenticator and E-Auth. However, the combination of password and authenticator made it distinguished from them. We especially stress this character of P-Auth as most of the existing authenticators require the use of asymmetric key pair that is hardly accepted by common people rather than human memorable password. Besides, though there are lots of works focus on the construction of password-based protocol design [10, 11, 12, 13] scarcely any of them makes contribution to authenticator. As we have to deal with the well known off-line dictionary attack faced by all password-based design, we adopt threshold scheme. Different from former works [14, 15, 16, 17], P-Auth needs no sever and it can be implemented by clients. In P-Auth quantity of partial secret send by one party to the other is less than threshold number. The receiver uses incept secrets and missing ones generated by him to reconstruct unite secret. Moreover, in order to strengthen this implementation, we acquire the notion of reputation that decides the substitution position of password in session key. Through introduction of these two notions, P-Auth also owns the character of dictionary attack resisting and forward security.

II. PRILIMINARIES

A. Authenticator

Model Here we put two security models together since the main difference between them is adversary's ability. Assume besides n parties $P_1 \cdots P_n$, there is an adversary A participates in the running of message-driven protocol π. A can control and schedule activations in π but is restricted to deliver message faithfully. This model is called an authenticated-link model (AM) and adversary A is named as AM-adversary. Different from AM, adversary in UM (unauthenticated-link model) owns an ability to determine the scheduling of events. He can delete, insert or modify messages in the communication channel at will. Capable adversary U like this is referred to as an UM-adversary.

Authenticator An authenticator is a transformation that takes a secure protocol in AM into an "equivalent" secure one in UM. Formally, it is defined as:

978-1-4244-1674-5/08 /$25.00 ©2008 IEEE

Definition A compiler C is an algorithm that takes for input descriptions of protocols and outputs descriptions of protocols. An authenticator is a compiler C where for any protocol π in AM, $\pi' \underset{\cdot}{=} \mathrm{C}\,(\pi)$ emulates π in UM.

Once the authenticator has been designed, one can construct an AM secure protocol and then implement the authenticator to produce an UM secure protocol. In [3] Bellare et al. construct two MT-authenticators which are one-flow protocol emulating that have to be applied on each message independently. In [9] Jiang et al. suggested authenticators that can overcome the round complexity problem on efficiency. However, as we have mentioned that these authenticators require asymmetric key system initial phrase and every party should maintain public/private key pairs that are not commonly used by people we designed a password-based authenticator that does not have this problem.

B. Password-based protocols

Although password-based protocols are efficient and easy to implement, it is insecure for the reason of low entropy. Assume the password is selected uniformly from a relatively small dictionary $D \subset \{0,1\}^n$, where $|\mathrm{D}| = \mathrm{poly}\,(\mathrm{n})$. An adversary can either choose to use on-line-guessing attack with success probability 1/|D| at each attempt or use off-line dictionary attack that performed by exhaustively enumerate all possible candidates to break the password. Therefore, the central challenge of this kind of protocol is how to prevent dictionary attack (while on-line attack can be prevented by presenting appropriate intervals between invalid trials [16]). In our password-based authenticator we introduced threshold scheme and "reputation" to avoid this problem.

C. Threshold scheme

Threshold scheme, also called secret sharing scheme, was introduced firstly in [18]. A (*k*, *n*) threshold scheme is a protocol among n players in which some dealer distributes partial information of a secret to n participants in order to meet the following statements. One is that any group which contains less than k participants can not obtain any information about the secret; the other is that any group of at least k participants can reconstruct the secret in polynomial time [17]. Common use of threshold scheme on password is distributing partial password to n different parties, if arbitrary k parties can recover the secret then these parties can be authenticated by the one who segments the secret. However, we do not implement threshold scheme directly on password in our authenticator. Instead, we occupy it with the auxiliary parameter "reputation". Besides, threshold scheme in our authenticator only held between two parties. As we only interest in the theoretical research on constructing the authenticator here computing cost stems from the employment of threshold scheme is neglected in this paper.

D. Reputation

In real world, people's behavior toward the ones they encounter is mostly based on their reputation. If it is good, one can make interaction with its holder without much scruple. Otherwise, one has to think twice or even refuse interchange. Derived from this phenomenon, designers of e-commerce (especially in P2P systems) import the conception of reputation. That is, as a summarized history of other people's transaction is provided by reputation system users can make use of it on deciding to what extent they should trust an unknown people before they themselves communicate with him/her [19]. In our work, "reputation" is not a real factor to decide whether one party should or should not trust another. Instead, it is an auxiliary element in unite secret construction in P-Auth. By using it, our authenticator occupies the character of forward security.

III. P-AUTH

In this section, we propose our password-based authenticator P-Auth. It makes use of modified E-Auth [8].

A. Notations

- pw : Password between communicators.
- L_{sk} : Session key length.
- L_{pw} : Password length.
- $\Delta(|\Delta| = L_{pw})$: One unit of interval in **unite secret**.
- $N = \lfloor L_{sk} / L_{pw} \rfloor$: Number of interval.
- R_{ij}^m : Reputation of party P_j' in P_i' 's view after m sessions. $R_{ij}^m \in \{1, 2, \cdots, N\}$.
- SR_{ij}^m : Reputation of party P_i' in P_j' 's view thought by P_i' . That is, it is a value guessed by P_i' , not the real value determined by P_j' . $SR_{ij}^m \in \{1, 2, \cdots, N\}$
- v : Shift rate on Δ between communicators.
- U : **Unite secret** generation function.
- e_{ij}^m : **Unite secret** held by party P_i' at the m-th session.
- T : A (*k*, *n*) threshold scheme.
- $PE_{ij}^m(s)$: The s-th **output secret** from party P_i' to P_j' on the m-th session ($1 \le s \le n$).
- $T_+PE_{ij}^m(s)$: Test secret at party P_i' 's side for **input secret** from party P_j' ($1 \le s \le n$).
- $T\_PE_{ij}^m(s)$: Test secret at party P_i' 's side for **input secret** from party P_j' ($1 \le s \le n$).
- e_{sk}^m : Session key.
- MAC : Message authentication code with a key space.
- N_i : Party P_i' 's MAC key. $N_i \leftarrow K$.
- E_{ij}^m : Using e_{ij}^m as the encryption key.

- $MAC < N_i >$: Message authentication code using key N_i.

B. The authenticator

Let π be any protocol. Assume $P_1, \cdots, P_n$ are n parties. Let $m_1, m_2, \cdots$ represent message flows exchanged between P_i and P_j in π. Without loss of generality, suppose m_1 is sent from P_i to P_j. Let $\pi' =$ P-Auth(π). We use P_i' to denote party i in protocol π' . Before applying our authenticator, parameters pw , L_{sk} , L_{pw} , R_{ij}^0 , R_{ji}^0 , v must be negotiated through secure channel. We also assume that e_{sk}^0 is already existed. Figure.1 shows the detail of our authenticator.

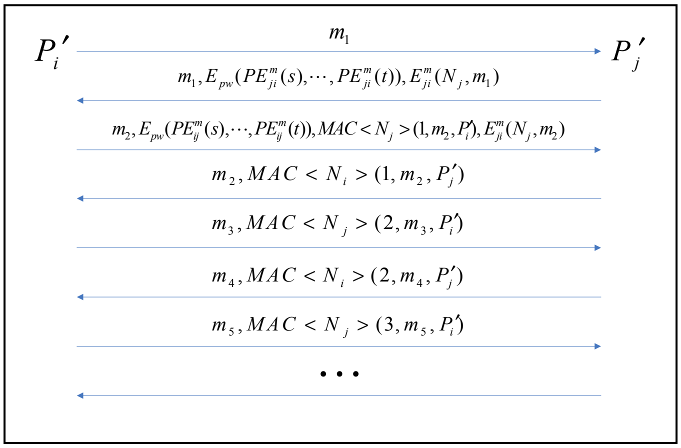
P_i'
P_j'
m_1
$m_1, E_{pw}(PE_{ji}^m(s), \cdots, PE_{ji}^m(t)), E_{ji}^m(N_j, m_1)$
$m_2, E_{pw}(PE_{ij}^m(s), \cdots, PE_{ij}^m(t)), MAC < N_j > (1, m_2, P_i'), E_{ji}^m(N_j, m_2)$
$m_2, MAC < N_i > (1, m_2, P_j')$
$m_3, MAC < N_j > (2, m_3, P_i')$
$m_4, MAC < N_i > (2, m_4, P_j')$
$m_5, MAC < N_j > (3, m_5, P_i')$
$\cdots$

Figure 1. A Password-based authenticator P-Auth

(1) Stage 1

(A) At P_i' 's side

- P_i' inputs e_{sk}^m , R_{ij}^m and pw to function U to generate his unite secret e_{ij}^m , one he uses to interact with P_j' .

$$e_{ij}^{m+1} = U(e_{sk}^m, R_{ij}^m, pw) \qquad (1)$$

Function U can be described as follows: Since $N = \lfloor L_{sk} / L_{pw} \rfloor$ is fixed, e_{sk}^m can be divided into N parts (, we neglect the remaining $L_{sk} \bmod L_{pw}$ part of e_{sk}^m). Then, we view R_{ij}^m as a pointer (or index). It shows us which part in e_{sk}^m should be replaced by the password. Therefore, after the alternation between part of e_{sk}^m and password, P_i' 's **unite secret** in the m-th session e_{ij}^m is generated.

- Using e_{ij}^m as input, threshold scheme T output n partial **output secret** $PE_{ij}^m(1), \cdots, PE_{ij}^m(n)$.

- (Guessing) Because P_i' has never been told by P_j' what his reputation is in P_j' 's view except R_{ji}^0 (value of $R_{ji}^m, m \geq 1$, never appears in any message flow). He has to guess his current reputation based on his former value and $v\Delta$ in each session. Such uncertain value is denoted as SR_{ij}^m . Using SR_{ij}^m instead of R_{ij}^m , P_i' repeats the first two steps under two different assumptions and then obtains two series of test secret.

$$\begin{cases} T_+PE_{ij}^m(1), \cdots, T_+PE_{ij}^m(n), when, SR_{ij}^m = R_{ji}^{m-1} + v\Delta \bmod N \\ T\_PE_{ij}^m(1), \cdots, T\_PE_{ij}^m(n), when, SR_{ij}^m = R_{ji}^{m-1} - v\Delta \bmod N \end{cases} \qquad (2)$$

(B) P_j' does the same work as P_i' .

(2) Stage 2

- In π' , P_i' first sends m_1 to P_j' .

- P_j' takes $N_j \leftarrow K$, responds with message m_1 , $E_{pw}(PE_{ji}^m(s), \cdots, PE_{ji}^m(t))$ ($k-1$ partial input secret randomly chosen from output secret $PE_{ji}^m(1), \cdots, PE_{ji}^m(n)$) and $E_{ji}^m(N_j, m_1)$.

- (Checking) On receiving P_j' 's answer, P_i' makes two tests. One is P_i' randomly chooses a test secret from $T_+PE_{ij}^m(1), \cdots, T_+PE_{ij}^m(n)$, the other is from $T\_PE_{ij}^m(1), \cdots, T\_PE_{ij}^m(n)$, to check whether one of them, together with input secret $PE_{ji}^m(s), \cdots, PE_{ji}^m(t)$ can reconstruct unite secret e_{ji}^m .Should P_i' succeed, he could decrypt $E_{ji}^m(N_j, m_1)$ to get N_j , and then he knows his current reputation R_{ij}^m in P_j' 's mind. The interaction goes on. If no, P_i' rejects any further message from P_j' .

  Next, P_i' takes $N_i \leftarrow K$, sends m_2 , $E_{pw}(PE_{ij}^m(s), \cdots, PE_{ij}^m(t))$ ($k-1$ **input secret** randomly chosen from $PE_{ij}^m(1), \cdots, PE_{ij}^m(n)$), $MAC < N_j > (1, m_2, P_i')$ and $E_{ji}^m(N_j, m_2)$ to P_j' .

- Similarly, P_j' reconstructs e_{ij}^m to verify P_i' 's identity and gets N_j for later use.

  Then, P_j' sends $m_2, MAC < N_i > (1, m_2, P_j')$ to P_i' .

Communicators will enter next stage if both of them do not reject each other after stage 2.

(3) Stage 3

If P_i' and P_j' send message alternately, their message should be

$m_i, MAC < N_j > (2i-1, m_i, P_i')$ and $m_j, MAC < N_i > (2i, m_j, P_j')$ respectively.

(4) Stage 4

After the m-th session is completed, P_i' arbitrarily set P_j''s new reputation in the following way:

$$R_{ij}^{m+1} = R_{ij}^{m} + v\Delta, or, R_{ij}^{m} - v\Delta \tag{3}$$

Also does P_j'.

C. Discussion and Analysis

From the description we can know that compared with E-Auth which only increases the round complexity by 4 based on MT-authenticators, P-Auth does not make any augment. Furthermore, as a password-based authenticator, P-Auth has the ability to resist off-line dictionary attack. Because parameter "reputation", which is used as a substitution index, has never been transmitted in any message flow at any form (except in the initial negotiation phrase) and it helps password to be combined into stronger session key to form unite secret, a higher entropy one. This operation magnifies adversary's load on exhaustive searching. Besides, as only participants know the other side's initial reputation and their shifting units per session ($v\Delta$) they are able to authenticate each other without any interaction of these two "password parameter" through the implementation of threshold scheme as we have mentioned that a (*k*, *n*) threshold scheme guarantees that only under the condition which partial secret's number exceeds $k-1$ can the original secret be recovered. In P-Auth, one participant only sends message flows contain $k-1$ partial secret and leave the other side patch the missing one to reconstruct secret through a "guessing and checking" way. The "guessing and checking" method is derived from the idea that: Suppose *A*, *B* and *C* are playing games. At first *A* lets *B* know that she has draw a circle on a paper. Then *A* hides the paper and adds an irregular figure on it, *B* knows the shape of this figure but not confirm at one or two corner. Later *A* tears the paper up, randomly chooses some fragments and take them away. Leaving the remaining pieces for *B* to recover what the specific figure is. *C* knows nothing from the beginning to the end, except that *A* draws two figures on a paper. If *C* wants to do the recovery work instead of *B* he will find it cost him heavy energy. However, as *B* knows all the possibilities he only needs to produce some candidate pictures, tear them up in the same way like *A* and then fix pieces into *A*'s fragmentary picture. If they are inosculated, *B* can affirm *A*'s graph. So if adversary does not get previous session key he will have no inputs to generate unite secret even if he obtained password. If adversary does not capture current reputation and $v\Delta$ between participants still he can not generate unite secret even if he held previous session key for the reason that if N is large enough the index number "reputation" itself can be viewed as a pseudorandom one. In this way, P-Auth realizes forward security.

We omit comparison between P-Auth and other ones in this paper since there are few works endeavor in the constructing of authenticator at present and most of them are derived from the original MT-authenticator. Formal proof on the security of P-Auth (provable security) is the further research in our work.

Acknowledgment

This work is supported by China National Natural Science Foundation under Grants No. 60673075 and China National High Technology Research and Development Program (863) under Grants No.2006AA01Z428.

REFERENCES

[1] R. Needham and M. Schroeder. Using Encryption for Authentication in Large Networks of Computers. Communications of the ACM, 21(12), December 1978.

[2] J. Clark and J. Jacob. A Survey of Authentication Protocol Literature: Version 1.0 17 November 1997.

[3] M. Bellare, R. Canetti, and H. Krawczyk, a modular approach to the design and analysis of authentication and key exchange protocols, STOC' 98, pp. 419-428.

[4] R. Canetti and H. Krawczyk, analysis of key-exchange protocols and their use for building secure channels, Advances in Cryptology-EUROCRYPT 2001, B. Pfitzmann (Ed.), LNCS 2045, Springer-Verlag, pp. 453-474, 2001.

[5] C. Boyd, W. Mao, K. Paterson, Key Agreement Using Statically Keyed Authenticators, ACNS 2004, M. Jakobsson et al. (Eds.), LNCS 3809, Spinger-Verlag, pp.248-262, 2004.

[6] Y. Tin, H. Vasanta, C. Boyd and J. Nieto, Protocols with Security Proofs for Mobile Applications, ACISP 2004, H.Wang et al. (Eds.), LNCS 3108, Springer-Verlag, pp. 358-369, 2004.

[7] M.Raimondo and R.Gennaro, New Approaches for Deniable Authentication, IACR eprint 2005/046. Available at http://eprint.iacr.org/2005/046.

[8] S. Jiang and G. Gong, Efficient Authenticators with Application to Key Exchange D. Won and S. Kim (Eds.): ICISC 2005, LNCS 3935, pp. 81–91, 2006.

[9] S. Haber, B. Pinkas, Securely Combining Public-Key Cryptosystems, CCS'01, Philadelphia, Pennsylvania, USA, November 5-8, 2001.

[10] M. Bellare, D. Pointcheval and P. Rogaway, authenticated key exchange secure against dictionary attacks, Advances in Cryptology-EUROCRYPT 2000, B. Preneel (Ed.), LNCS 1807, Springer-Verlag, pp. 139-155,2000.

[11] J. Katz, R. Ostrovsky and M. Yung, efficient password-authenticated key exchange using human-memorable passwords, Advances in Cryptology-EUROCRYPT 2001, B. Pfitzmann (Ed.), LNCS 2045, Springer-Verlag, pp. 475-494, 2001.

[12] O. Goldreich and Y. Lindell, session-key generation using human passwords only, Advances in Cryptology-CRYPTO'01, J. Kilian (Ed.), LNCS 2139, Springer-Verlag, pp. 408-432, 2001.

[13] R. Gennaro and Y. Lindell, a framework for password-based authenticated key exchange, Advances in Cryptology-EUROCRYPT 2003, E. Biham (Ed.), LNCS 2656, Springer-Verlag, pp. 524-543, 2003.

[14] Y. Hitchcock, Y. Shing T. Tin, J. M. G. Nieto, C. Boyd, and P. Montague, A Password-Based Authenticator: Security Proof and Applications, INDOCRYPT 2003, pp. 388-401.

[15] S. Shin, K. Kobara, and H. Imai, A Secure Threshold Anonymous Password-Authenticated Key Exchange Protocol A. Miyaji, H. Kikuchi, and K. Rannenberg (Eds.): IWSEC 2007, LNCS 4752, pp. 444–458, 2007.

[16] Viet, D.Q., Yamamura, A., Tanaka, H.: Anonymous Password-Based Authenticated Key Exchange. In: Maitra, S., Madhavan, C.E.V., Venkatesan, R. (eds.) INDOCRYPT 2005. LNCS, vol. 3797, pp. 244–257, 2005.

[17] S. Lee, K. Han, S. Kang, K. Kim, and S. Ine, Threshold Password-Based Authentication Using Bilinear Pairings, S.K. Katsikas et al. (Eds.): EuroPKI 2004, LNCS 3093, pp. 350–363, 2004.

[18] A.Shamir, How to Share a Secret, Communication of the ACM, Vol.22, No.11, pp.612-613, Nov.1979.

[19] S. Marti, Trust and reputation in peer-to-peer networks, a dissertation submitted to the department of computer science and the committee on graduate studies of stanford university in partial fulfillment of the requirements for the degree of doctor of philosophy, may 2005.