Addendum to Summary of AMP

Taekyoung Kwon

Sejong University, Seoul 143-747, Korea
tkwon@sejong.ac.kr

Abstract. This document corroborates the AMP protocols by making some minor modification to the previous work. So the final versions of AMP are described for November 2003 discussion by the IEEE P1363 Standard Working Group. The contribution includes AMP and TP-AMP protocols.

1 Introduction

User authentication is necessary for the typical case that a human being resides as a client and tries to log on to a remote server machine. The server must be able to determine the user's identity reliably over a public or private channel. Password authentication is one of such methods, in which simply the user memorizes a password while the server maintains a user profile that associates the user name and the password verifying information. The intrinsic problem with this method is the password, associated with each user, has low entropy, so that it is not easy to protect the password information against the notorious password guessing attacks by which attackers search the relatively small space of human-memorable passwords.

Since a pioneering method that resists the password guessing attacks was introduced to cryptographic protocol developers [20], there has been a great deal of work for password authenticated key agreement on the framework of Diffie-Hellman [9], for example, EKE [5] and so on [26, 27]. Readers are referred to [12] for complete references. Compared to the typical authenticated key agreement, the password-based schemes are totally expensive due to the low entropy of human-memorable passwords, specifically in the augmented model which was contrived to resist server compromise. Provable security makes the schemes even harder to be practical. From the theoretical perspective, several methods that are much more expensive but provably secure in the standard model, have been presented [10, 15]. From the practical perspective, the practice-oriented security models are applied for examining the security of protocols [1–3]. For example, EKE2 and AuthA are provably secure in both the random oracle and ideal cipher models [3, 4, 8], while PAK and PAK-Z (that improves the efficiency of PAK-X by specifying a general function for a digital signature) are provably secure in the random oracle model [7, 21, 22]. In spite that those practice-oriented models are not standard paradigms, they could give strong evidences that the target schemes are not flawed.

At present, SPEKE [13], SRP [30], PAK [22], and AMP [17] are being discussed by the IEEE P1363 Standard Working Group as practical protocols for standardization on password-based public key cryptographic techniques [11]. PAK and SPEKE are known as the most acceptable 'three-pass' protocols to the standard body, while AMP and SRP are regarded as the most suitable 'four-pass' protocols. The work of IEEE P1363.2 is impressive and valuable in many aspects; for instance, a new attack called the 'two-for-one' guessing attack[1] against the four-pass protocols was found and resolved in the process [11, 29].

[1] An active attacker can validate two password guesses in one impersonation attempt. The first attack against SRP was discovered by D. Bleichenbacher in 2000, while the similar attack on AMP was caught by M. Scott [29]. However, both protocols were fixed to resist respective attacks in the IEEE P1363.2 process.

1.1 This Document

This document describes the final versions of AMP protocols, as an addendum to the summary of AMP [18]. So we make some minor modification to the related protocols. The main contribution includes AMP (Figure 1) and TP-AMP (Figure 2) where AMP stands for Authentication and key agreement via Memorable Passwords and TP-AMP for Three-Pass AMP.

1.2 Preliminaries

First we follow the formal description of [3]. Let *Clients* and *Servers* be the finite, disjoint, nonempty sets of principals which are modeled as probabilistic polynomial time algorithms with input/output tapes. A client $C \in Clients$ has a secret password π which is drawn randomly from small space Π of size N. A server $S \in Servers$ has a transformed-password τ_C which contains an entry per client, where $\tau_C \stackrel{\mathrm{T}}{\leftarrow} \pi$ for C. We call π and τ_C long-lived-weak keys (LL-keys). They are the same values in the balanced model, while not in the augmented model. For convenience, S is assumed as an IP address of the server machine while C is a user name.

A client and a server should agree on algebraic parameters related to Diffie-Hellman key agreement. Let κ and ℓ denote security parameters. Notice that κ is regarded as the general parameter for hash functions and secret keys (say 160 bits), while ℓ is thought of as the special parameter for public keys (say 1024 or 2048 bits). Let q of size at least κ and p of size ℓ be primes such that $p = rq+1$ for some value r co-prime to q. Let g be a generator of $\mathbb{G}_q$ where $\mathbb{G}_q$ is a q-order subgroup of a multiplicative group $\mathbb{Z}_p^*$. So we can define $\bar{\mathbb{G}}_q = \{g^x \bmod p | x \in \mathbb{Z}_q^*\}$ where $|\bar{\mathbb{G}}_q| = q - 1$. Let us often omit 'mod p' from the expressions that are obvious in $\mathbb{Z}_p^*$. In spite that we still recommend to use a *secure prime* such that each factor of r except 2 is of size at least κ or even a *safe prime* such that $r = 2R$ for a prime R [17, 19, 30], the final versions of AMP are flexible in allowing any other form of a prime p. Let $\{0,1\}^*$ denote the set of finite binary strings and $\{0,1\}^n$ the set of binary strings of length n. Then we could have random oracles denoted by h_i: $\{0,1\}^* \rightarrow \{0,1\}^\kappa$ or H_i: $\{0,1\}^* \rightarrow \{0,1\}^\ell$. Let $\mathsf{ACCEPTABLE}(\cdot)$ denote an acceptable function which may return true if its pre-image satisfies the given security properties. For more details of the legacy protocols, readers are referred to the previous work [7, 11, 17, 21, 22].

2 AMP Protocols

AMP stands for the Authentication and key agreement via Memorable Passwords [17], and was contributed to the IEEE P1363.2 standard work. On standardization, the protocol was improved in order for resisting the newly found attack called two-for-one guessing and for minimizing the necessary computation. Also a three-pass derivative called TP-AMP (Three-Pass AMP) was proposed. We corroborate those AMP protocols briefly.

2.1 Four-Pass AMP Protocol

AMP was basically designed as a four-pass protocol in the augmented model [17]. Figure 1 depicts the final version of AMP protocol that are contributed to the IEEE P1363.2 Standard Working Group. This protocol is a slightly improved and general version of [18] since a secure or safe prime p was strongly assumed in the previous protocol. The difference between them

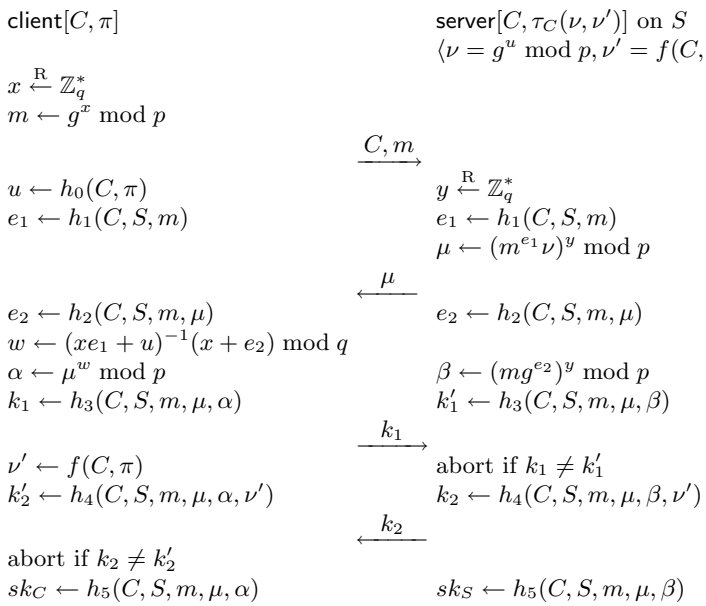

Fig. 1. AMP (Authentication and key agreement via Memorable Passwords)

is to inject ν' into k_2 so that the authenticity of server can strongly be confirmed for the case that p is neither a secure prime nor a safe prime. Note that $\nu' = f(C, \pi)$ where $f(\cdot)$ is a strong one-way hash function. As a result, on the registration phase, a user chooses a name C and a password π while the server S saves user's profile $\langle C, \tau_C \rangle$ in its stable storage where $u = h_0(C, \pi)$, $\nu = g^u$, $\nu' = f(C, \pi)$ and $\tau_C = \langle \nu, \nu' \rangle$. So it might be easier for the client to check the validity of μ (say, free from small subgroup confinement of μ) than the previous protocol because k_2 shows the authenticity of server regardless of the property of μ. However, we can remove ν' from the above protocol when we choose p as a secure prime or a safe prime. Both AMP2 and AMP3 protocols can enjoy this modification as well. Refer to [18] for them.

2.2 Three-Pass AMP Protocol

We also corroborate the final version of TP-AMP by defining a unique protocol. TP-AMP stands for the Three-Pass Authenticated key agreement via Memorable Passwords for practical password authenticated key agreement. We describe the protocol in more detail.

On the registration phase, a user chooses a name C and a password π while the server S saves user's profile $\langle C, \tau_C \rangle$ in its stable storage where $\gamma = H_0(C, \pi)$, $\gamma' = \gamma^{-1}$, $u = h_1(C, \pi)$, $\nu = g^u$, and $\tau_C = \langle \gamma', \nu \rangle$. For convenience, S is assumed as an IP address of the server machine. Figure 2 depicts the protocol and we explain it step by step.

On the protocol run, the user types C and π into the client machine. Then the client chooses x at random from $\mathbb{Z}_q^*$, and computes γ in order to obtain $m = g^x\gamma$. The client sends a commitment message $\langle C, m \rangle$ to the server.

$$1.\ C \rightarrow S : C, g^x\gamma \bmod p \tag{1}$$

After or before sending message 1, the client could compute γ' and the user's *amplified password* such that $w = u^{-1}(x + e) \bmod q$ by obtaining $u = h_1(C, \pi)$ and $e = h_2(C, S, m)$, and keeps them while waiting for message 2.

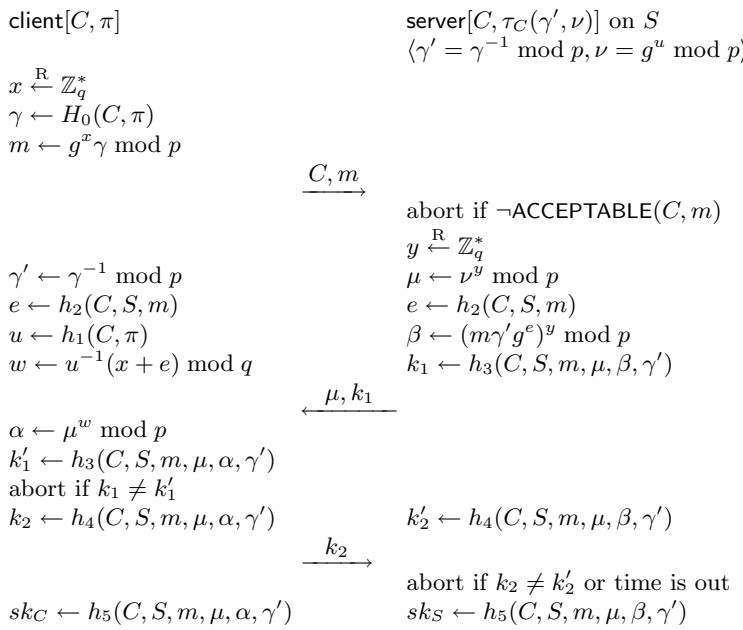

Fig. 2. TP-AMP (Three-Pass AMP Protocol)

Upon receiving message 1, the server should abort it if ACCEPTABLE(C, m) returns false. Otherwise, the server fetches $\langle C, \tau_C \rangle$ from its storage and chooses y at random from $\mathbb{Z}_q^*$ in order to obtain $\mu = \nu^y$. Then the server computes $e = h_2(C, S, m)$, $\beta \equiv (m\gamma' g^e)^y \equiv g^{(x+e)y} (\mathrm{mod} p)$ and $k_1 = h_3(C, S, m, \mu, \beta, \gamma')$, and sends a challenge message $\langle \mu, k_1 \rangle$ to the client.

$$2.\ S \rightarrow C : \nu^y \bmod p, h_3(C, S, m, \mu, \beta, \gamma') \tag{2}$$

After or before sending message 2, the server could compute $k_2' \leftarrow h_4(C, S, m, \mu, \beta, \gamma')$ and keeps it while waiting for message 3.

On receiving message 2, the client raises μ to the amplified password so that $\alpha \equiv \mu^w \equiv g^{y(x+e)} (\mathrm{mod} p)$, and computes $k_1' = h_3(C, S, m, \mu, \alpha, \gamma')$. If k_1 is not equal to k_1', the client should abort this session. Otherwise, the client computes $k_2 = h_4(C, S, m, \mu, \beta, \gamma')$ and sends a response message k_2 to the server.

$$3.\ C \rightarrow S : h_4(C, S, m, \mu, \alpha, \gamma') \tag{3}$$

After or before sending message 3, the client could compute a session key such that $sk_C = h_5(C, S, m, \mu, \alpha, \gamma')$ and deletes any other ephemeral values. Also it aborts if time is out.

Upon receiving message 3, the server should abort this session if k_2 is not equal to k_2'. Otherwise, the server should compute a session key such that $sk_S = h_5(C, S, m, \mu, \beta, \gamma')$ and deletes any other ephemeral values.

As a result, the client and the server could authenticate each other via the memorable password and agree on the same session key $sk_C (= sk_S)$ because $\alpha \equiv \beta \equiv g^{(x+e)y} (\mathrm{mod} p)$.

We need to define the special function called ACCEPTABLE$(\cdot)$ since the server should abort when it returns false upon receiving $\langle C, m \rangle$. The function follows:

ACCEPTABLE$(\cdot)$

INPUT: $\langle C, m \rangle$

OUTPUT: Return *false* if C is being served by another instance;
else if the failure count of C is greater than or equal to its limit δ;
else if $m \notin Z_p^*$; */
Return *true* otherwise;

The reason for checking whether C is being served by another instance is to resist the so-called *many-for-many guessing*[2] attack [18].

3 Conclusion

This document is an addendum to the summary of AMP [18] for November 2003 discussion by the IEEE P1363 Standard Working Group.

References

1. M. Bellare and P. Rogaway, "Entity authentication and key distribution," In *Crypto 1993*, LNCS 773, pp.232-249, 1993.
2. M. Bellare and P. Rogaway, "Provably secure session key distribution-the three party case," In *ACM Symposium on the Theory of Computing*, pp.232-249, 1993.
3. M. Bellare, D. Pointcheval and P. Rogaway, "Authenticated key exchange secure against dictionary attack," In *Eurocrypt 2000*, LNCS 1807, pp.139-155, 2000.
4. M. Bellare and P. Rogaway, "The AuthA protocol for password-based authenticated key exchange," Submission to the IEEE P1363.2 study group, available from `http://www.cs.ucdavis.edu/ rogaway/papers/autha.ps`
5. S. Bellovin and M. Merritt, "Encrypted key exchange : password-based protocols secure against dictionary attacks," In *IEEE Symposium on Research in Security and Privacy*, pp. 72-84, 1992.
6. S. Bellovin and M. Merritt, "Augmented encrypted key exchange: a password-based protocol secure against dictionary attacks and password-file compromise," In *ACM Conference on Computer and Communications Security*, pp. 244-250, 1993.
7. V. Boyko, P. MacKenzie and S. Patel, "Provably secure password authenticated key exchange using Diffie-Hellman," In *Eurocrypt 2000*, LNCS 1807, pp.156-171, 2000.
8. E. Bresson, O. Chevassut, and D. Pointcheval, "Security proofs for an efficient password-based key exchange," In *ACM Conference on Computer Communications Security*, 2003.
9. W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol.22, no.6, pp.644-654, November 1976.
10. O. Goldreich and Y. Lindell, "Session-Key Generation Using Human Passwords Only," In *Cypto 2001*, LNCS 2139, pp.408-432, 2001.
11. IEEE P1363-2, "Standard specifications for password-based public key cryptographic techniques," available from `http://grouper.ieee.org/groups/1363/`, December 2002.
12. Phoenix Technologies, Inc., "Research Papers on Strong Password Authentication," available from `http://www.integritysciences.com/links.html`, 2002.
13. D. Jablon, "Strong password-only authenticated key exchange," *ACM Computer Communications Review*, vol.26, no.5, pp.5-26, 1996.

[2] The author has found that an active attacker can validate as many password guesses as (s)he makes server instances invoked and this can be a real attack against every three-pass password protocol. For example, a multi-threaded or multi-processed server is vulnerable to this attack. However, this attack is negligible in the four-pass protocols since the server does not give sufficient information to the adversary forward and the client is usually not capable of listening to so many concurrent requests in the opposite case. Also the well-known predecessors, EKE [5] and A-EKE [6], avoid this attack very *impressively* by not optimizing the protocol steps. The author's suggestion is to check whether the client is being served by another server instance for resisting this attack.

14. D. Jablon, "Extended password key exchange protocols," In *WETICE Workshop on Enterprise Security*, pp.248-255, 1997.
15. J. Katz, R. Ostrovsky, and M. Yung, "Efficient Password-Authenticated Key Exchange Using Human-Memorable Passwords ," In *Eurocrypt 2001*, LNCS 2045, pp.475-494, 2001.
16. T. Kwon and J. Song, "Secure agreement scheme for g^{xy} via password authentication," *Electronics Letters*, vol.35, no.11, pp.892-893, 27th May 1999.
17. T. Kwon, "Authentication and key agreement via memorable password," In *ISOC Network and Distributed System Security Symposium*, February 2001.
18. T. Kwon, "Summary of AMP," `http://grouper.ieee.org/groups/1363/passwdPK/contributions/ampsummary.pdf` or `http://dasan.sejong.ac.kr/ tkwon/amp.html`, August 2003.
19. C. Lim and P. Lee, "A key recovery attack on discrete log-based schemes using a prime order subgroup," In *CRYPTO 97*, pp.249-263, 1997.
20. M. Lomas, L. Gong, J. Saltzer, and R. Needham, "Reducing risks from poorly chosen keys," In *ACM Symposium on Operating System Principles*, pp.14-18, 1989.
21. P. MacKenzie, "More efficient password-authenticated key exchange," In *RSA Conference*, Cryptographers Track, LNCS 2020, pp.361-377, 2001.
22. P. MacKenzie, "The PAK suite: Protocols for Password-Authenticated Key Exchange," Submission to IEEE P1363.2, April 2002.
23. A. Menezes, P. van Oorschot and S. Vanstone, *Handbook of applied cryptography*, CRC Press,Inc., pp.517-518, 1997.
24. P. van Oorschot and M. Wiener, "On Diffie-Hellman key agreement with short exponents," In *Eurocrypt 96*, pp. 332-343, 1996.
25. S. Patel, "Number theoretic attacks on secure password schemes," In *IEEE Symposium on Security and Privacy*, 1997.
26. R. Perlman and C. Kaufman, "PDM: A new strong password-based protocol," In *USENIX Security Symposium*, pp.313-321, 2001.
27. M. Roe, B. Christianson and D. Wheeler, "Secure sessions from weak secrets," Technical report from University of Cambridge and University of Hertfordshire, 1998, available from `http://www.ccsr.cam.ac.uk/techreports/tr4/index.html`
28. C. Schnorr, "Efficient identification and signatures for smart cards," In *CRYPTO 89*, pp.239-251, 1989.
29. M. Scott, *Personal communication*, July 2001.
30. T. Wu, "Secure remote password protocol," In *ISOC Network and Distributed System Security Symposium*, 1998.
31. T. Wu, "SRP6: Improvements and refinements to the secure remote password protocol," Unpublished document, October 2002.