Revision of AMP in IEEE P1363.2 and ISO/IEC 11770-4

Taekyoung Kwon

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

Abstract. While preparing a full document of AMP (Authenticated key agreement via Memorable Passwords), we have found a small but significant security problem in the current standardizing version of IEEE P1363.2 and ISO/IEC 11770-4. This document describes the problem and proposes the substitution of the former proposal AMP+ for resolving it.

1 Introduction

AMP stands for Authenticated key agreement via Memorable Passwords and is a cryptographic protocol designed for strong password-based authentication and key agreement in a distributed environment [5]. There are a large number of cryptographic protocols having the same goal, and among them several protocols including AMP are being standardized by IEEE P1363 Standard Working Group and ISO/IEC SC27 Standardization Committee, respectively, from the practical perspectives [2].

AMP was first introduced to IEEE P1363 in May 2000 and presented at ISOC's Network and Distributed System Security Symposium in February 2001. While designing AMP in the early stage, practical security was mainly considered in the augmented model. In other words, we pursued the highest computational performance and simplicity of the protocol without losing its security. Several variants including AMP+ and amplified password file were manipulated from the same standpoint in order to add more security functions [5].

In the progressive work of IEEE P1363.2, the so-called 'two-for-one' guessing attack[1] was formulated against the four-pass protocols such as SRP and AMP [2, 8]. As a response to the corresponding call in 2001, in order to resist this attack, we proposed to replace the original proposal with AMP+ that is first described in [5] and further reformulated as the final version of AMP [6, 7]. While AMP+ being examined, it was reconsidered that AMP+ needs slightly more computation for a server than the first proposal does. Thus, we revised it from the practical viewpoint again, and the resulting protocol replaced the first proposal in 2002. This is the current standardizing version of AMP in both IEEE P1363.2 and ISO/IEC 11770-4. However, the additional needs for checking parameters were pointed out by the committee, and we felt AMP+ is better than the current standardizing version from many aspects. Subsequently, we posted the summaries of AMP in 2003 and proposed to restore AMP+ as a standardizing version of AMP [6, 7], while it has not been consented yet.

1.1 This Document

Recently, we have found a small but significant security problem in the current standardizing version. This document describes the problem and proposes to replace the current version with AMP+ [5], the final version of AMP, or the unified variant AMP2 [6].

[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 [8]. However, both protocols were fixed to resist respective attacks in the IEEE P1363.2 process.

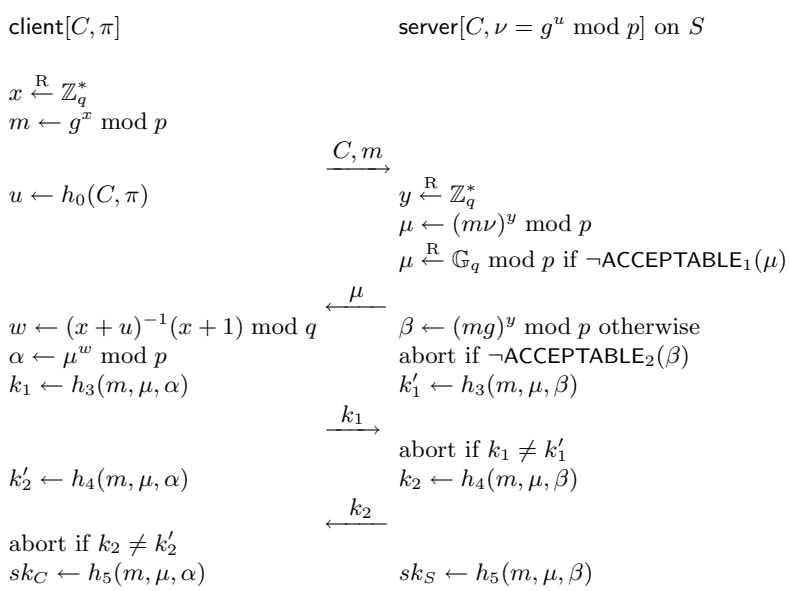

Fig. 1. Current standardizing version of AMP

1.2 Preliminaries

Readers are referred to [5] and [6] for various versions of AMP, and [2] and [3] for standardization. In this document, we follow the formal description of [1]. A client C and a server S 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 $\mathbb{Z}_p^*$. We strongly recommend to use a *secure prime*[2] such that each factor of r except 2 is of the size at least κ or a *safe prime* such that $r = 2R$ for a prime R [5, 9]. 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: $\{0,1\}^* \to \{0,1\}^\kappa$ or H_i: $\{0,1\}^* \to \{0,1\}^\ell$. 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 [4].

2 The Problem

Figure 1 depicts the current standardizing version of AMP in IEEE P1363.2 and ISO/IEC 11770-4. The function $\mathsf{ACCEPTABLE}_i(\cdot)$ may return false if the order of input is unacceptably

[2] It was our mistake to consider a random prime in [7] but the need of secure or safe prime was already discussed in the related teleconference. As for every AMP, a secure or safe prime is necessary but we strongly recommend the secure prime since the safe prime makes it slightly heavier to compute an inverse in $\mathbb{Z}_q^*$.

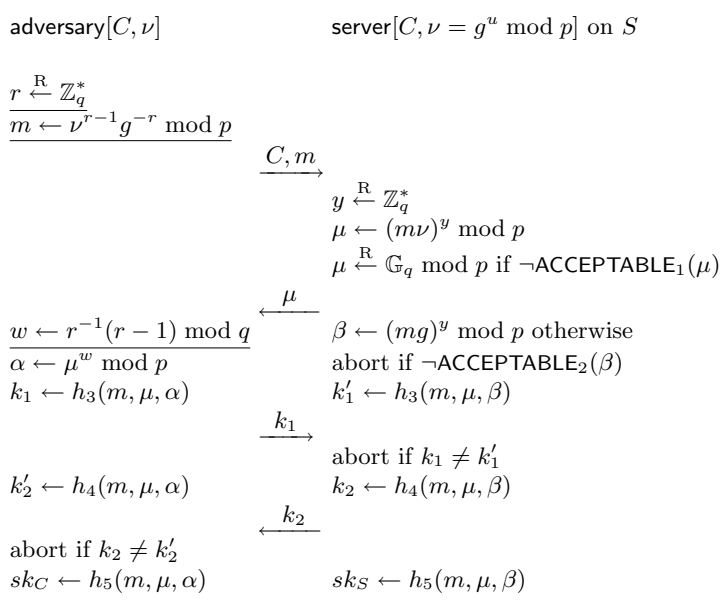

Fig. 2. Server compromise attack to the current standardizing version

small, so as to resist two-for-one guessing attacks. However, as we mentioned already, we feel that AMP+ is better than this standardizing version [6]. The main reasons are; (1) eventually a specific manipulation is necessary for ACCEPTABLE$_i(\cdot)$ checking small orders and (2) a timing attack[3] is to be considered for two-for-one guessing when receiving μ from the server. In addition to these minor problems, we have recently found that the current standardizing version has the more significant security problem.

Figure 2 depicts that an adversary is able to impersonate the target client without knowing the exact password if the server is once compromised. In other words, a server compromise attack is possible even though the protocol is designed in the augmented model. The attack is quite simple and the adversarial operations are underlined in Figure 2. The adversary who obtained ν, sets m as $\nu^{r-1}g^{-r}$ in $\mathbb{G}_q$ by choosing r at random, and sends it to the server. Upon receiving μ from the server, the adversary then sets w as $r^{-1}(r-1)$ in $\mathbb{Z}_q$. That's it. The resulting secret value is $\alpha = \beta = \nu^{(r-1)}g^{-(r-1)}$ in $\mathbb{G}_q$, while the server couldn't be aware of this attack since r makes m look random to the server.

3 The Solution

As for the attack above, the weak point of the current standardizing version is that the adversary is able to determine the value w when choosing r such that $w = r^{-1}(r-1)$. Since it is possible for the adversary who does not have a correct password or u, to compute w even before manufacturing m, the server cannot prevent it. In order to resist this attack, it is necessary to randomize w against the adversary choosing r, so that (s)he cannot determine it in advance.

[3] An attacker who poses as a client can measure a time till receiving μ because a random subgroup element must *additionally* be selected for μ unless ACCEPTABLE$_1(\cdot)$ returns true. So, we suggest to compute $c = m\nu$ first, and send $\mu = c^y$ if ACCEPTABLE$_1(\cdot)$ returns true and $\mu = g^y$ otherwise.

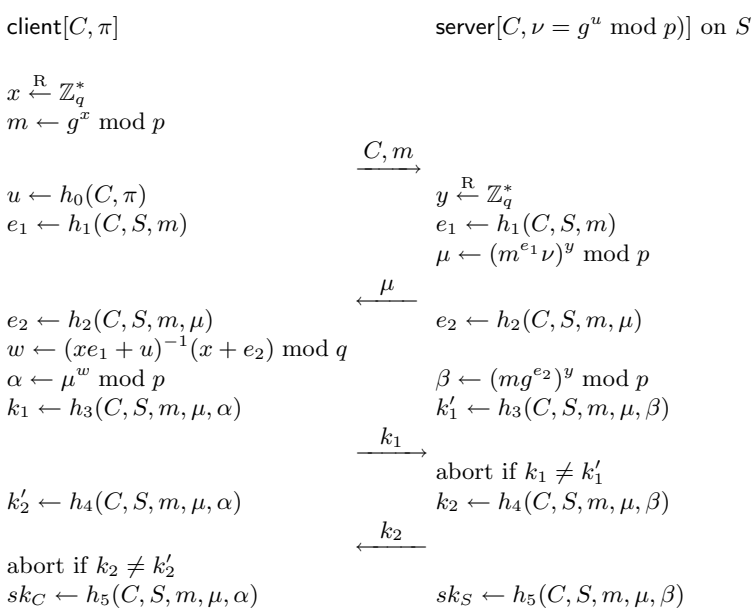

Fig. 3. AMP+, the final version of AMP

Fortunately, we already have at least two secure versions of AMP that are not far from the current standardizing version with regard to the protocol formulation. One is AMP+ [5] that is the final version of AMP, and the other is the unified variant AMP2 [6]. Both protocols make w random in the way that the server raises m to a random oracle for computing μ, for example, $\mu = (m^e\nu)^y$ where $e = h_1(C, S, m)$. In other words, the adversary is unable to determine w on choosing r due to the random oracle. Figure 3 depicts AMP+, the final version of AMP [5, 6], while Figure 4 shows AMP2 [6].

AMP+ raises m to e_1 for computing μ and g to e_2 for computing β where $e_1 = h_1(C, S, m)$ and $e_2 = h_2(C, S, m, \mu)$, respectively [5, 6]. The length of e_1 must be κ at least, while that of e_2 can be defined as 40 bits practically. Note that the first proposal of AMP raises g to the random oracle for computing β [5], while the current standardizing version removes it.

AMP2 unifies the use of random oracle for exponents and the removal of exponentiating g [6], and is slightly more efficient than AMP+. Both protocols are secure against the new server compromise attack. They also resist the two-for-one guessing attack without checking the small order of element as the current standardizing version does.

We believe both protocols can replace the current standardizing version of AMP, so as to remove several security concerns such as server compromise, small order checking, and timing attack. As a result, the highest computational performance and simplicity without losing security, that is the spirit of AMP, can be achieved in the standardization. We would like to comment that TP-AMP was also designed in the same standpoint [7].

4 Conclusion

The final version of AMP (AMP+) or the unified version AMP2 must replace the current standardizing version of AMP in IEEE P1363.2 and ISO/IEC 11770-4 because of the weakness against server compromise and previous security concerns on the current version.

$\mathsf{user}[id, \pi]$ on A		$\mathsf{server}[id, \nu = g^u \bmod p]$ on B
$x \leftarrow_R \mathbb{Z}_q$		
$m \leftarrow g^x \bmod p$		
	$\xrightarrow{C, m}$	
$u \leftarrow h_0(C, \pi)$		$y \leftarrow_R \mathbb{Z}_q$
$e \leftarrow h_1(C, S, m)$		$e \leftarrow h_1(C, S, m)$
		$m' \leftarrow m^e \bmod p$
$w \leftarrow (xe + u)^{-1}(x + 1) \bmod q$		$\mu \leftarrow (m'\nu)^y \bmod p$
	$\xleftarrow{\mu}$	
$\alpha \leftarrow \mu^w \bmod p$		$\beta \leftarrow (m'g)^y \bmod p$
$k_1 \leftarrow h_3(C, S, m, \mu, \alpha)$		$k_1' \leftarrow h_3(C, S, m, \mu, \beta)$
	$\xrightarrow{k_1}$	
		abort if $k_1 \neq k_1'$
$k_2' \leftarrow h_4(C, S, m, \mu, \alpha)$		$k_2 \leftarrow h_4(C, S, m, \mu, \beta)$
	$\xleftarrow{k_2}$	
abort if $k_2 \neq k_2'$		
$K \leftarrow h_5(C, S, m, \mu, \alpha)$		$K' \leftarrow h_5(C, S, m, \mu, \beta)$

Fig. 4. AMP2

References

1. M. Bellare, D. Pointcheval and P. Rogaway, "Authenticated key exchange secure against dictionary attack," In *Eurocrypt 2000*, LNCS 1807, pp.139-155, 2000.
2. IEEE P1363-2, Standard specifications for password-based public key cryptographic techniques, available from `http://grouper.ieee.org/groups/1363/`, December 2002.
3. ISO/IEC SC27 FCD 11770-4, Information technology - Security techniques - Key management - Part 4: Mechanisms based on weak secrets, 2005
4. D. Jablon, Research Papers on Strong Password Authentication, available from `http://www.jablon.org/passwordlinks.html`, 2004.
5. T. Kwon, "Authentication and key agreement via memorable password," In *ISOC Network and Distributed System Security Symposium*, February 2001.
6. 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.
7. T. Kwon, "Addendum to Summary of AMP," `http://grouper.ieee.org/groups/1363/passwdPK/contributions/ampsummary2.pdf` or `http://dasan.sejong.ac.kr/ tkwon/amp.html`, November 2003.
8. M. Scott, *Personal communication*, July 2001.
9. T. Wu, "SRP6: Improvements and refinements to the secure remote password protocol," Unpublished document, October 2002.