ISSN NO. 2320-5407

Journal homepage: http://www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH

RESEARCH ARTICLE

Cryptanalysis of a Known Key Exchange Password Scheme

Sattar J Aboud
Iraqi Council of Representatives, Department of Information Technology, Baghdad-Iraq.

Manuscript Info

Manuscript History:

Received: 11 August 2013
Final Accepted: 24 August 2013
Published Online: September 2013

Abstract

In this paper we analyzed the Harkins scheme that is the password authenticated key exchange scheme which has been proposed for an Internet as the standard scheme for universal internet use. We studied a security of this scheme and developed the attack that is able of obtaining both the session key and password from the real participant. The attack then applied and investigates the achievement to decide a time-gauge needed to effectively done an attack.

Introduction

In 2012 Harkins proposed a password authenticated key exchange scheme for exchanging session keys by common authentication in networks [1]. Recently, Harkins presented the modified of a scheme to an Internet as the standard scheme for universal Internet use. We see that both alternatives are basically a same scheme; however some application facts are dissimilar. It is claimed that a scheme is anti-active attack, passive attack, and offline dictionary attack [2]. But, in fact no security proofs are provided to verify a claim. The absence of security proofs has produced some doubts between the Internet participants. Though, no entity has offered actual attacks [3].

In this paper, we study the security characteristics of the Harkins scheme. Also, we will demonstrate that the scheme is an alternative to the offline dictionary attack. We will base the analysis on an original description of the scheme. Though, an attack we will introduce is slightly appropriate to an alternative one.

The Harkins Scheme

Harkins scheme is relied on discrete logarithm assumption. This means that the uses of Harkins scheme can either use processes on the elliptic curve or the finite field [4]. No assumptions are created regarding a basic group, rather than a calculation of discrete logarithm problem which is intractable for a level of security needed. In every case, there are two processes which can be achieved, a factor process that accepts input of two values and yields the third value, and the scalar process that accepts input of a factor and the scalar and yields a third value.

Suppose we use a finite field as an instance. Assume p be the prime number. We indicate the finite cyclic group by f which is the subgroup of Z_p with prime order q where $q / p-1$. We indicate a member process by $u \cdot v$ for member u and v. These representations are consistent with those working by the finite field. The Harkins scheme operates as follows:

1. Entity A and entity B have the common password from which every can deterministically produce the password $w \in f$.
2. Entity A arbitrarily selects two numbers $i_A, j_A \in q$, computes $y_A = i_A + j_A \bmod q$ and $e_A = w^{-j_A} \bmod p$ then posts y_A, e_A to entity B .
3. Entity B arbitrarily selects two numbers $i_B, i_m \in q$, computes $y_B = i_B + j_B \bmod q$ and $e_B = w^{-j_B} \bmod p$ then posts y_B, e_B to entity A .

4. Entity A computes a common secret $b = (w^{y_B} e_B)^{i_A} = w^{i_A i_B} \bmod p$
5. Entity B computes a common secret $b = (w^{y_A} e_A)^{i_B} \bmod p$
6. Entity A passes $u = h(b \mid e_A \mid y_A \mid e_B \mid y_B)$ to entity B such that h is a secure hash function
7. Entity B posts $v = h(b \mid e_B \mid y_B \mid e_A \mid y_A)$ to entity A
8. Entity A and entity B verify that hashes values are true and then they generate the common key $d = h(b \mid e_A \cdot e_B \mid (y_A + y_B) \bmod q)$.

Attack on Harkins Scheme

In this section we will discuss the following subsections regarding the attack on Harkins scheme.

3.1 Attack Approach

The Harkins claimed that the scheme is anti-offline dictionary attacks. But, no security proof is provided. In its place, an author gives the experiential security analysis as follows.

1. Entity A has the common password from which can deterministically produce the password $w \in f$.
2. Entity A arbitrarily selects two numbers $i_A, j_A \in q$, computes $y_A = i_A + j_A \bmod q$ and $e_A = w^{-j_A} \bmod p$ then posts y_A, e_A to entity B.
3. The hacker entity B arbitrarily selects $y_B \in q$, and finds $e_B = t_n$ then posts y_B, e_B to entity A.
4. Entity A computes a common secret $b = (w^{y_B} e_B)^{i_A} = w^{i y_B i_A} t_n^{i_A} \bmod p$
5. Entity A passes $u = h(b \mid e_A \mid y_A \mid e_B \mid y_B)$ to entity B
6. The hacker entity B posts (w, v, d) by offline search algorithm (u, y_B, e_B) to entity A
7. Entity A verify that $v = h(b \mid e_B \mid y_B \mid e_A \mid y_A)$
8. Entity A computes verify the common key $d = h(b \mid e_A \cdot e_B \mid (y_A + y_B) \bmod q)$.

It is supposed that an active hacker will choice a random integer for j_B and find $e_B = g^{j_B}$ with g is a group generator for f. Then, a hacker will receive hash result for which only unidentified input to a hash function is z with $w = g^z$ so, for the offline dictionary attack is successful, a hacker will be able to find z for the arbitrary value in f which is difficult to calculate. We indicate that calculating $e_B = g^{j_B}$ does not the best choice to the active hacker. In its place, a hacker can use an algorithm 1. Thus, a hacker finds $e_B = t_n$ with t_n is a primitive of the small subgroup Z_p of order n. Then, a common secret calculated by entity A that $b = (w^{y_B} t_n)^{i_A} = w^{y_B i_A} t_n^{i_A}$, this is only unknown by which a hash value post by entity A is dependent. A hacker then applies Algorithm 1:

1. To get a victim password w
2. To forge the valid reply v to avoid authentication
3. To find secrecy of communication by getting a session key d.

This attack is feasible as t_n creates the small subgroup and a password that adequately authorize dictionary attack. When $u = u^-$ in algorithm 1 is true, we have $b = b^-$ since a hash is the random oracle. Therefore, we find $w^{y_B i_A} t_n^{i_A} = (w^{-y_A} e_A)^{y_B} c_x$ with c_x is an unknown small subgroup component.

Algorithm 1: Offline Search Algorithm

Input: u, y_B, e_B

Output: w, v, d

for every w^- *in dictionary do*

{ *for every* c_x *in subgroup do*

{ $b^- = (w^{-y_A} e_A)^{y_B} c_x$

$u^- = h(b^- \mid e_A \mid y_A \mid e_B \mid y_B)$

$if\ \ u = u^{-}\ then\ \{$

$w = w^{-}$

$v = h(b^{-} \mid e_B \mid y_B \mid e_A \mid y_A)$

$d = h(b^{-} e_A \cdot e_B \mid (y_A + y_B) \bmod q)$

$return(w, v, d)\ \}\ \}\ \}$

When re-arranging the terms, we get: $\frac{w^{y_B i_A}}{(w^{-y_A} e_A)^{y_B}} = \frac{c_x}{t_n^{i_A}}$

Remark: a term on a left side is the factor in the subgroup of prime order q where as a term on a right side is the factor in the small subgroup of order n. As, $q \neq n$ an equality have only if both sides are identity factors in Z_p^* that is 1. So $(w^{-y_A} e_A)^{y_B} = w^{i_A y_B}$, in which only probable value for w^{-} is $w^{-} = p$. When successfully getting a victim password, a hacker is capable to forge the valid reply and pass it back to entity A, thus entity A is ignorant that the password has been compromised [5]. Finally, a hacker can get a common session key and involve with entity A in a following secret communication relied on session key.

Results

We started by employing the brute force method to find prime factors of $p-1$with p is a prime number with 1024 bit size. In a trial, we searched for prime factors of length less than 32 bit size. Thus, we obtained the following prime factors: 2, 3, 13, 23 and 463907. Then, we computed the generators for every related small subgroup and achieved the set of tests to fix a time to finish the offline dictionary attack for every subgroup. Every set of trials included rising an attack with dictionaries of 1000, 10000 and 100000 arbitrary password elements. The different dictionary length allowed to gauge the growth in dictionary length will affect a time taken to finish an attack. The time gauged was a time to attempt teach possible password, till a right password was found. Every trial was achieved30 times. We observe that only one probable password was found in each trial and this was a password selected by a true participant. The time spends to verify all passwords in the subgroup of length 463907 and dictionary list are shown in Table 1.

Table 1: The Subgroup of Length 463907

Dictionary Length	Mean Time to Attempt All Passwords (MS)	Std Dev
1000	16894596	146432
10000	169475631	4527603
100000 169338910072654427		

This shows that there is the linear relationship between dictionary length and a time spend to attempt all passwords. The times spend to verify all possible passwords with the dictionary length of 1000 as a subgroup length differs and are illustrated in Table 2.

Table 2: The Dictionary Length of 1000

Subgroup length	Mean Time to Attempt All Passwords (MS)	Std Dev
3	5657	73
13	6323	84
23	7702	189

In all instances the trials were run under Windows 7 Home premium on the 2.53GHz PC with 6 GB of RAM memory and 64bit Operating System. We observe that selected of the times gauged are adequately large that entity A might end a protocol because of a large time spent for a hacker to reply. But, there are three mitigating issues to describe:

1. We have gauged a mean time to attempt all passwords; actually we will expect a hacker to catch a right password without attempting all passwords.
2. The hacker will have the resources to allocate the computations over some high execution computers, decreasing a computation time considerably.
3. When a protocol is ended, a hacker will determined a password and can be capable to utilize it in another run of a protocol.

Discussion

Small subgroup attacks can be stopped by verifying that a received a member E is the element of a group being used by a public key scheme. This can be done by verifying that E is element of a super-group, that E is not an identity member and that E^q is congregant to an identity member. The value of this verifies known as a public key validation in key exchange protocol emphasized by [6,7].But, to validate the public key will need the full exponentiation over a finite group, which will considerably reduce the scheme efficiency and cause it less attractive compare with its competitors. Therefore, it stays arguable within a cryptography public when a public key validation is vital. However, for the case of a Harkins scheme, we have illustrated that an absence of public key validation reduces a scheme totally insecure.

Conclusion

We have illustrated that a Harkins scheme is vulnerable under the small subgroup typed offline dictionary attack. This attack can be avoided by adding the public key validation, which will reduce scheme efficiency. In the past years, numerous key exchange schemes have neglected public key validation, but are thus discovered vulnerable to small subgroup attacks regardless of an increased efficiency. Harkins is the example of this attack and will not be a last.

References

[1] Dan Harkins, "Simultaneous authentication of equals: A secure, password-based key exchange for mesh networks", In Sensor Technologies and Applications, 2008. SENSORCOMM '08, Second International Conference, pp. 839–844, Aug. 2008.

[2] Dan Harkins, "Dragonfly key exchange-internet research task force internet draft", http://tools.ietf.org/html/draft-irtf-cfrg-dragonfly-00accessed Jan 2013, 2012.

[3] Kaufman, C., Hoffman, P., Nir, Y., and P. "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

[4] McGrew, D., Igoe, K., and M. Salter, "Fundamental Elliptic Curve Cryptography Algorithms", RFC 6090, February 2011.

[5] Barker, E., Johnson, D., and M. Smid, "Recommendations for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography", NIST Special Publication 800-56A, March 2007.

[6] Alfred Menezes and Berkant Ustaoglu, "On the importance of public-key validation in the mqv and hmqv, key agreement protocols", In Proceedings of the 7th international conference on Cryptology in India, INDOCRYPT'06, pp. 133–147, Berlin, Springer, 2006.

[7] Clancy, T. and H. Tschofenig, "Extensible Authentication Protocol-Generalized Pre-Shared Key (EAP-GPSK) Method", RFC 5433, February 2009.