An Attack on the *Interlock Protocol* When Used for Authentication

Steven M. Bellovin*
Michael Merritt†

February 4, 1993

Abstract

Exponential key exchange may be used to establish secure communications between two parties who do not share a private key. It fails in the presence of an active wiretap, however. Davies and Price suggest use of Shamir and Rivest's "Interlock Protocol" to surmount this difficulty. We demonstrate that an active attacker can, at the cost of a timeout alarm, bypass the password exchange, and capture the passwords used. Furthermore, if the attack is from a terminal or workstation attempting to contact a computer, the attacker will have access before any alarm can be sounded.

Keywords: cryptography, protocol, security, authentication, exponential key exchange, Diffie-Hellman

*smb@research.att.com, AT&T Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974.

†mischu@research.att.com, AT&T Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974.

I Introduction

The exponential key exchange protocol[1] has been suggested as a form of public-key cryptosystem. It is also useful if two parties wish to set up a secret conversation without prior arrangement, as the public keys are relatively easy to generate.

The dialog works as follows. Let α and β be large, publicly-known numbers. Suppose that A wishes to talk privately with B. Each side picks a random number, A_R and B_R. The following messages are sent:

$$\begin{array}{cccc} \boxed{A} & & & \boxed{B} \\ \alpha^{A_R} \bmod \beta & \longrightarrow & & \\ & \longleftarrow & \alpha^{B_R} \bmod \beta. & \end{array}$$

At this point, A, who knows A_R, can calculate

$$(\alpha^{B_R})^{A_R} \bmod \beta \equiv \alpha^{A_R B_R} \bmod \beta.$$

B can perform a similar calculation to obtain the shared key $\alpha^{A_R B_R} \bmod \beta$. An intruder cannot do the same, however. For example, in order to calculate A_R from $\alpha^{A_R} \bmod \beta$, the eavesdropper must calculate the logarithm of $\alpha^{A_R} \bmod \beta$ in the field $GF(\beta)$. This is believed to be difficult, though notable progress has been made of late, especially for small values of β [2].

II Active Wiretaps

As elegant as it is, exponential key exchange is vulnerable to active, "man in the middle" attacks. That is, suppose an intruder Z can not only listen to messages between A and B, but can also modify messages, delete messages, or even generate totally new ones. In that case, Z can imitate B when talking to A, and imitate A when talking to B:

$$\begin{array}{ccccc} \boxed{A} & & \boxed{Z} & & \boxed{B} \\ \alpha^{A_R} \bmod \beta & \longrightarrow & & & \\ & & \alpha^{Z_R} \bmod \beta & \longrightarrow & \\ & & & \longleftarrow & \alpha^{B_R} \bmod \beta \\ & \longleftarrow & \alpha^{Z'_R} \bmod \beta. & & \end{array}$$

At this point, A and Z can compute

$$(\alpha^{A_R})^{Z'_R} \bmod \beta \equiv (\alpha^{Z'_R})^{A_R} \bmod \beta \equiv \alpha^{A_R Z'_R} \bmod \beta$$

while Z and B can compute

$$(\alpha^{Z_R})^{B_R} \bmod \beta \equiv (\alpha^{B_R})^{Z_R} \bmod \beta \equiv \alpha^{Z_R B_R} \bmod \beta.$$

Let $E_{A,B}(M)$ represent an encryption of M using the private key derived from the exponential key exchange. For simplicity, we assume that the entire message is encrypted as a single block. Now, suppose A sends a password P_A to B, encrypted as $E_{A,Z'}(P_A)$. Then Z can intercept and decrypt the message, obtain P_A, re-encrypt it with $\alpha^{Z_R B_R} \bmod \beta$ and send it on its way:

$$\begin{array}{lcccl} \boxed{A} & & \boxed{Z} & & \boxed{B} \\ E_{A,Z'}(P_A) & \longrightarrow & & & \\ & & E_{Z,B}(P_A) & \longrightarrow & \\ & & & \longleftarrow & E_{Z,B}(P_B) \\ & \longleftarrow & E_{A,Z'}(P_B). & & \end{array}$$

Here, B has received $E_{Z,B}(P_A)$, decrypted and matched it against the stored password for A, and responded with its own password, P_B; this, in turn, is decrypted and re-encrypted by Z for transmission to A.

Continuing in this way, Z may eavesdrop on the entire conversation between A and B, or substitute new messages to either recipient. Neither A nor B knows that Z is intruding.

Davies and Price [3, page 222] describe a scheme by Shamir and Rivest that is meant to surmount this problem if A and B share passwords P_A and P_B. A and B encrypt their passwords using that key, producing $E_{A,B}(P_A)$ and $E_{A,B}(P_B)$. Rather than transmitting those quantities intact, each party divides the encrypted password into two halves, producing $E_{A,B}(P_A)\langle 1\rangle$, $E_{A,B}(P_A)\langle 2\rangle$, $E_{A,B}(P_B)\langle 1\rangle$, and $E_{A,B}(P_B)\langle 2\rangle$. They then exchange halves alternately:

$$\begin{array}{lcl} \boxed{A} & & \boxed{B} \\ E_{A,B}(P_A)\langle 1\rangle & \longrightarrow & \\ & \longleftarrow & E_{A,B}(P_B)\langle 1\rangle \\ E_{A,B}(P_A)\langle 2\rangle & \longrightarrow & \\ & \longleftarrow & E_{A,B}(P_B)\langle 2\rangle. \end{array}$$

Suppose, now, that Z is interfering in this dialog. In that case, A is really sending $(E_{A,Z'}(P_A))\langle 1\rangle$ in the first message. But Z cannot decrypt $E_{A,B}(P_A)\langle 1\rangle$ until $E_{A,B}(P_A)\langle 2\rangle$ arrives, and thus cannot re-encrypt it using $E_{Z,B}$. Thus, A and B can detect the intrusion.

III The Attack

Assume that A represents a user of computer B, and that it is such access that Z wishes to obtain. We demonstrate that the above algorithm still allows Z to gain access to B.

Assume that Z is sitting in the middle of the dialog shown above. Since Z

does not know P_B, half of some arbitrary encryption is sent instead.

$\boxed{A}$		$\boxed{Z}$	$\boxed{B}$
$(E_{A,Z'}(P_A))\langle 1\rangle$	$\longrightarrow$		
	$\longleftarrow$	$(E_{A,Z'}(P_?))\langle 1\rangle$	
$(E_{A,Z'}(P_A))\langle 2\rangle$	$\longrightarrow$		

At this point, Z can combine the two halves of A's encrypted password P_A and — knowing $\alpha^{A_R Z'_R} \bmod \beta$ — can decrypt it, and continue the dialog with B:

$\boxed{A}$	$\boxed{Z}$		$\boxed{B}$
	$(E_{Z,B}(P_A))\langle 1\rangle$	$\longrightarrow$	
		$\longleftarrow$	$(E_{Z,B}(P_B))\langle 1\rangle$
	$(E_{Z,B}(P_A))\langle 2\rangle$	$\longrightarrow$	
		$\longleftarrow$	$(E_{Z,B}(P_B))\langle 2\rangle$.

That is, Z has completed A's half of the authentication procedure, as far as B can tell.

The connection from Z to A may be dropped at this point. True artistry demands Z should do this by simulating line noise or a network failure. It does not matter much, however; Z has full access to B before A can sound any alarm. (Recall that we are assuming that Z has full control of the communications channel between A and B. If there is an auxiliary, unauthenticated channel between the two parties, A might use it to yell for help. But care should be taken, as false messages on that channel might be used for a denial of service attack.) If A attempts to reconnect, Z can either continue the simulation of a communications problem, or it can complete the authentication dialog. Note that P_A and P_B are compromised at this point. If A attempts to reconnect, Z can either continue the simulation of a communications problem, or it can complete the authentication dialog with A, using P_B.

An obvious counter is to modify the dialog slightly so that B speaks first. The user password P_A is much more sensitive than the host password P_B; it would make sense for A to be assured that it was speaking to the genuine B before transmitting a password. This, too, is insecure; Z can still break in to B, though possibly for a shorter time before an alarm is raised.

The attack this time requires that Z set up a second, parallel connection to B, using exponents Z'' and B'. (In fact, it does not matter what the exponents are here; Z and B will go through a separate exponential key exchange to calculate them.) This second session is used to capture P_B:

$\boxed{Z}$		$\boxed{B}$
	$\longleftarrow$	$(E_{Z'',B'}(P_B))\langle 1\rangle$
$(E_{Z'',B'}(P_?))\langle 1\rangle$	$\longrightarrow$	
	$\longleftarrow$	$(E_{Z'',B'}(P_B))\langle 2\rangle$.

Naturally, Z refrains from sending the second half of $P_?$ at this time. Instead, the captured P_B is used on the original connection to spoof A.

$\boxed{A}$		$\boxed{Z}$
	$\longleftarrow$	$(E_{A,Z'}(P_B))\langle 1\rangle$
$(E_{A,Z'}(P_A))\langle 1\rangle$	$\longrightarrow$	
	$\longleftarrow$	$(E_{A,Z'}(P_B))\langle 2\rangle$
$(E_{A,Z'}(P_A))\langle 2\rangle$	$\longrightarrow$	

At this point, Z has captured P_A as well. This may be used to complete the authentication dialog with B. Presumably, at some point B will get suspicious that the dialog on the second channel has not been completed; even if a communications problem is simulated, security alarms will likely be raised. It is not likely, however, that A will be locked out instantly; as noted earlier, automatic lockouts can be used for denial of service.

If two channels into B are not available, the same scheme works using successive calls from Z to B; in that case, however, an alarm will be sounded before Z has completed the second connection. The scheme originally outlined provides a period before the alarm, when the intruder has access to the system.

IV Alternatives

Davies has suggested[4] that validation of the passwords be done in parts. That is, since A knows P_B and the negotiated encryption key, she also knows the two halves of the encrypted message, and can validate them as they arrive. Thus, when Z sends $(E_{A,Z'}(P_?))\langle 1\rangle$, it is apparent that the message does not contain an encryption of P_B, so the reply of $(E_{A,Z'}(P_A))\langle 2\rangle$ will never be sent.

This strategy foils our attack. However, this requires that the unencrypted messages contain, in addition to passwords, only information known to both parties. In particular, nonce fields, unpredictable timestamps or random padding cannot be used. Furthermore, it is common practice to avoid storing cleartext versions of others' passwords; rather, a one-way hash of the password is generally kept[5]. In this case, the encrypted password can only be verified when both halves are received; half of the encrypted password is useless.

A different authentication protocol based on exponential key exchange is given in [6]. Briefly, a shared secret key is used to encrypt the exponentials before transmission. In addition to providing both authentication and secrecy, this scheme guards against password-guessing attacks[5, 7, 8, 9]. Further, it can be used with most public key cryptosystems, not just exponential key exchange. It suffers from the same flaw as does Davies' defense: it is not suitable for situations where one-way hashing of passwords is used.

Another alternative is presented in [10]. Günther presents an authentication system that is also based on the discrete logarithm problem; in it, the user's identity is encoded into the public key. Thus, the secrecy negotiation protocol

itself yields the user's identity. However, this scheme requires that the user store a bulky secret key, rather than a simple password.

V A Lesson

The original man-in-the-middle attack is countered by the protocol attributed to Rivest and Shamir. As we have seen, that protocol in turn falls to a slightly more complicated attack. Instead of considering *specific* attacks in designing protocols, it may be much more fruitful to consider *classes* of attacks embodied in specific adversarial assumptions [11, 12]. Certainly, no protocol should be accepted purely on the basis of its defeating a single attack.

References

[1] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. IT-11, pp. 644–654, November 1976.

[2] B. A. LaMacchia and A. M. Odlyzko, "Computation of discrete logarithms in prime fields," *Designs, Codes, and Cryptography*, vol. 1, pp. 46–62, 1991.

[3] D. W. Davies and W. L. Price, *Security for Computer Networks*. John Wiley & Sons, second ed., 1989.

[4] D. Davies, July 1990. Private conversation.

[5] R. H. Morris and K. Thompson., "Unix password security," *Communications of the ACM*, vol. 22, p. 594, November 1979.

[6] S. M. Bellovin and M. Merritt, "Encrypted key exchange: Password-based protocols secure against dictionary attacks," in *Proc. IEEE Computer Society Symposium on Research in Security and Privacy*, (Oakland), pp. 72–84, May 1992.

[7] F. T. Grampp and R. H. Morris, "Unix operating system security," *AT&T Bell Laboratories Technical Journal*, vol. 63, pp. 1649–1672, October 1984.

[8] D. V. Klein, ""Foiling the cracker": A survey of, and improvements to, password security," in *Proceedings of the USENIX UNIX Security Workshop*, (Portland), pp. 5–14, August 1990.

[9] P. Leong and C. Tham, "Unix password encryption considered insecure," in *Proc. Winter USENIX Conference*, (Dallas), 1991.

[10] C. G. Günther, "An identity-based key-exchange protocol," in *Advances in Cryptology: Eurocrypt '89*, pp. 29–37, Springer-Verlag, 1990.

[11] R. DeMillo and M. Merritt, "Protocols for data security," *Computer*, vol. 16, pp. 39–50, February 1983.

[12] J. Moore, "Protocol failures in cryptosystems," *Procedings of the IEEE*, vol. 76, pp. 594–602, May 1988.