An attack on some multi-party key agreement protocols

Kenneth G. Paterson,
Information Security Group,
Royal Holloway, University of London
Egham, Surrey, TW20 0EX, UK
kenny.paterson@rhul.ac.uk

Abstract

Lee, Lee and Lee [App. Math. and Comp. Vol. 159 , 2004, pp. 317-331] recently presented a collection of $n+1$ different n-party key agreement protocols based on multi-linear forms. Here we show that $n-2$ of the protocols are completely insecure, being vulnerable to impersonation attacks.

Keywords: Cryptanalysis; multi-linear forms; key agreement protocol

1 Introduction

Recently, Lee, Lee and Lee [4] presented a collection of $n+1$ different n-party key agreement protocols. Their protocols make use of multi-linear forms [3] and are based on the protocols developed by Al-Riyami and Paterson in [1, 2]. We show that, contrary to the claim of [4, p. 323], $n-2$ of the $n+1$ protocols are easily broken using an impersonation attack. In our attack, the adversary impersonates the parties in the protocol to one another and learns the session keys held by the parties at the end of the attack. Thus the protocols of [4] do not meet even the most basic security requirement for a key agreement protocol.

In Section 2 we review the protocols of [4], while in Section 3 we present our attacks on the protocols of [4].

2 The protocols of Lee, Lee and Lee

The protocols of [4] operate in the following mathematical setting. Groups G_1 and G_2 are of the same prime order (p say) and $e_n : G_1^{n-1} \to G_2$ is an $(n-1)$-multi-linear map, that is, a map satisfying:

$$e_{n-1}(x_1^{a_1}, x_2^{a_2}, \ldots, x_{n-1}^{a_{n-1}}) = e_{n-1}(x_1, x_2, \ldots, x_{n-1})^{a_1 a_2 \ldots a_{n-1}}$$

for any $a_1, a_2, \ldots, a_{n-1} \in \mathbb{Z}_p$ and any $x_1, x_2, \ldots, x_{n-1} \in G_1$. Such maps were introduced to cryptography by Boneh and Silverberg in [3]. However, it should be noted that no implementations of such maps suitable for use in cryptographic applications are currently known to exist.

In the protocols of [4], each of the n participants A_i has a long-term private key $x_i \in \mathbb{Z}_p$ and a corresponding public key g^{x_i} where g is a fixed generator of G_1. Each participant then picks at random $a_i \in \mathbb{Z}_p^*$, computes g^{a_i} and broadcasts this value to all the other participants. Then $n+1$ possible different methods for the protocol participants to agree on a common key are presented in [4]. Of these, two methods, namely MAK-A and MAK-C, need not concern us further here. The remaining methods are named MAK B-j ($1 \leq j \leq n-1$) in [4]. In method MAK B-j, the key computed by the participants is of the form:

$$K = \prod_{I \subset [n],\ |I|=j} e_{n-1}(g, g, \ldots, g)^{x_I a_{[n]\setminus I}}.$$

where $[n]$ denotes the set $\{1, 2, \ldots, n\}$, x_I denotes the product $\prod_{i \in I} x_i$ and $a_{[n]\setminus I}$ denotes the product $\prod_{i \in [n]\setminus I} a_i$. We note that a different (and arguably less clear) notation is used to describe these protocols in [4].

Thus the set of exponents appearing in the calculation of K consists of all $\binom{n}{j}$ possible expressions comprising a product of j distinct terms of the form x_i (which we call "x" terms) and $n-j$ distinct terms of the form a_i (which we call "a" terms). It is not hard to see how participant A_i, in possession of his private values x_i, a_i and all the public values g^{x_j}, g^{a_j} (for $j \neq i$) can exploit the multi-linear nature of the map e_{n-1} to compute K. For example, when $n = 4$ and $j = 2$, A_1 can compute the expression $e_3(g, g, g)^{x_2 x_4 a_1 a_3}$ appearing in the product for K as $e_3(g^{x_2}, g^{x_4}, g^{a_3})^{a_1}$, while A_2 can compute it as $e_3(g^{x_4}, g^{a_1}, g^{a_3})^{x_2}$.

3 Impersonation attack

We now demonstrate an impersonation attack on the protocols MAK B-j for $1 \leq j \leq n-2$, that is, on all the MAK B protocols of [4] with the

exception of MAK B-$(n-1)$.

Let j satisfy $1 \leq j \leq n-2$ and consider an adversary E who intercepts and replaces the values broadcast by the participants in protocol MAK B-j. For each k, adversary E intercepts the value g^{a_k} sent by participant k to the other protocol participants and replaces it with a value $g^{a'_k}$, where $a'_k \in \mathbb{Z}_p$ is known to E.

Consider how participant A_k in receipt of values $g^{a'_1}, g^{a'_2}, \ldots g^{a'_n}$ (excluding the term $g^{a'_k}$) calculates his key, K_k say. This key is a computed by A_k as a product of terms of the form $T'_I = e_{n-1}(g, g, \ldots, g)^{x_I a_{[n]\setminus I}}$ where I is a subset of $[n]$ of size j and the a_i are replaced by a'_i for each $i \neq k$.

We show how E can also compute each of these terms T'_I. Let $I = \{i_1, i_2, \ldots, i_j\}$. There are two cases. In the first case, we have $k \in I$. Then E knows all of the "a" terms appearing in the exponent of T'_I (they are all of the form a'_i) and can exploit the multi-linearity of e_{n-1} to compute:

$$T'_I = e_{n-1}(g^{x_{i_1}}, g^{x_{i_2}}, \ldots, g^{x_{i_j}}, g, \ldots, g)^{a'_{[n]\setminus I}}$$

where $a'_{[n]\setminus I} = \prod_{i \in [n]\setminus I} a'_i$. In the second case, we have $k \notin I$. Then E knows all of the "a" terms appearing in the exponent of T'_I with the exception of a_k and so can compute:

$$T'_I = e_{n-1}(g^{x_{i_1}}, g^{x_{i_2}}, \ldots, g^{x_{i_j}}, g^{a_k}, g, \ldots, g)^{a'_{[n]\setminus (I\cup\{k\})}}$$

where $a'_{[n]\setminus (I\cup\{k\})} = \prod_{i \in [n]\setminus (I\cup\{k\})} a'_i$. Here, we use the fact that $j \leq n-2$ to ensure that the $j+1$ different terms $x_{i_1}, x_{i_2}, \ldots, x_{i_j}, a_k$ can be moved from the exponent to "inside" the map e_{n-1}.

Thus it is apparent that E can also compute all the terms T'_I, and hence the key K_k held by participant k.

Example 1 *When $n = 4$ and $j = 2$, participant A_1 receives values $g^{a'_2}$, $g^{a'_3}$ and $g^{a'_4}$ from E. He computes the expression $e_3(g, g, g)^{x_2 x_4 a_1 a'_3}$ as part of his computation of K_1. The adversary E can also compute this expression using the formula:*

$$e_3(g, g, g)^{x_2 x_4 a_1 a'_3} = e_3(g^{x_2}, g^{x_4}, g^{a_1})^{a'_3}$$

and his knowledge of a'_3. In a similar fashion, E can calculate the 5 other expressions involved in A_1's computation of K_1.

Note that at the end of the attack, the various participants hold different keys, all of which are known to the adversary E. This is normal in

impersonation attacks. If the keys are subsequently used, for example, for encryption, then this allows E to act as a man-in-the-middle, intercepting, decrypting, reading and then re-encrypting any data sent by any participant to any other participant.

Acknowledgement

I am grateful to C.J. Mitchell for bringing [4] to my attention and to the anonymous reviewer for comments which helped to improve the presentation.

References

[1] S.S. Al-Riyami and K.G. Paterson. Tripartite Authenticated Key Agreement Protocols from Pairings. Cryptology ePrint Archive, Report 2002/035, `http://eprint.iacr.org/`, 2002.

[2] S.S. Al-Riyami and K.G. Paterson. Tripartite Authenticated Key Agreement Protocols from Pairings. In *K.G. Paterson (ed.), Proc. IMA Conference on Cryptography and Coding*, Lecture Notes in Computer Science Vol. 2898, pp.332-359, Springer-Verlag, Berlin, 2003.

[3] D. Boneh and A. Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics*, **324** (2003), 71-90.

[4] Y.-R. Lee, H.-S. Lee and H.-K. Lee. Multi-party authenticated key agreement protocols from multi-linear forms. *Applied Mathematics and Computation*, **159** (2004), 317-331.