# A Note on Security Protocol for Multicast Communications

Martin Stanek

Department of Computer Science, Comenius University
Mlynská dolina, 842 48 Bratislava, Slovak Republic
(Email: stanek@dcs.fmph.uniba.sk)
*(Received Sept. 8, 2010; revised and accepted Dec. 27, 2010)*

## Abstract

We show that the protocol recently proposed in [2] for securing multicast communication is completely insecure.

*Keywords: Backward secrecy, forward secrecy, Key distribution, multicast*

## 1 Introduction

Various applications use multicast communication for delivering data from a central sender to multiple receivers (members). The need to secure such communication usually reduces to the problem of cryptographic key distribution. A protocol for securing multicast communication should provide:

- Support for basic operations: join (new member joining the group of receivers) and leave (excluding a member from the group of receivers).
- Backward secrecy: the member cannot access multicast data sent before he/she joins the group.
- Forward secrecy: the member cannot access multicast data sent after he/she leaves the group.

Designing secure and efficient protocol (in terms of communication complexity or number of required cryptographic operations) is not an easy task and many different protocols were proposed, see for example [1, 3, 4, 5]. Recently, Aly Saroit, El-Zoghdy, and Matar [2] designed a protocol which is particulary efficient in case of leave operation. We show their protocol is completely insecure.

## 2 The Protocol and Its Insecurity

Members are organized in subgroups with dedicated server (called "subgroup controller" – SC). These subgroups can be viewed as independent multicast groups. Multicast is performed in two stages: first, distributing data to SC in each subgroup and then (after reencryption) SC sends data to the members of corresponding subgroup.

Let us note that concept of SC in the proposed protocol is rather inflexible, since (according [2]) these entities must be trusted by other members and they are not allowed to leave the multicast group.

The construction of keys in the protocol resembles Diffie-Hellman key exchange. The computation is performed in multiplicative subgroup of GF($p$) for sufficiently large prime number $p$. Let $g$ be a generator for this group. Let $P_{j,1}, \ldots, P_{j,m}$ be the members of $j$-th subgroup, and $S_j$ be the SC of this subgroup. The SC generates a secret $a_{j,i}$ for member $P_{j,i}$ (the secret is assigned but not given to $P_{j,i}$) satisfying natural conditions: $2 \le a_{j,i} \le p-2$, and $\gcd(a_{j,i}, p-1) = 1$. The member $P_{j,i}$ does not know this secret. However, (s)he knows the value $K_j^{(i)} = g^{a_{j,i}} \bmod p$ and the inverses of all secrets assigned to other members of the subgroup, i.e. $a_{j,k}^{-1} \bmod (p-1)$ for all $k \neq i$. The server $S_j$ has a secret $s_j$ satisfying condition $2 \le s_j \le p-2$.

**Remark.** *We have changed the notation of members' secrets, because of ambiguous notation $a_j^i$ used in the original paper, which looks like an exponentiation. Trivially, the semantics of $a_j^i$ cannot be exponentiation, since this gives all members immediate access to their own assigned secrets, e.g. $P_{j,2}$ knows the inverse $a_j^{-1}$, and can easily compute $a_j^2 \bmod (p-1)$. Knowing own assigned secret violates the forward secrecy as described later.*

The symmetric subgroup key $K_{SGj}$ for subgroup $j$ is given as (in fact, the computation is performed on a binary tree, but it is not relevant for our analysis):

$$K_{SGj} = g^{a_{j,1} \cdot \ldots \cdot a_{j,m} \cdot s_j} \bmod p.$$

When a new member $P_{j,m+1}$ joins the subgroup $j$, the following happens (beside other actions):

- SC generates new secret $s'_j$, i.e. the new subgroup key $K'_{SGj}$ (in order to guarantee backward secrecy)

will be

$$K'_{SGj} = g^{a_{j,1}\cdot\ldots\cdot a_{j,m}\cdot a_{j,m+1}\cdot s'_j} \bmod p.$$

- SC sends (besides other data) to the members of the subgroup following message encrypted by $K_{SGj}$: $\{K'_{SGj}, a^{-1}_{j,m+1}\}_{K_{SGj}}$. This ensures that old members receive new key $K'_{SGj}$, and knowing the inverse of $a_{j,m+1}$ allows efficient implementation of leave operation (which is the main contribution of [2]) by computing subgroup key by members as (in this case assuming the leaving member is again $P_{j,m+1}$):

$$\begin{aligned} & (K'_{SGj})^{a^{-1}_{j,m+1}} \bmod p \\ = \; & (g^{a_{j,1}\cdot\ldots\cdot a_{j,m}\cdot a_{j,m+1}\cdot s'_j})^{a^{-1}_{j,m+1}} \bmod p \\ = \; & g^{a_{j,1}\cdot\ldots\cdot a_{j,m}\cdot s'_j} \bmod p. \end{aligned}$$

- SC also sends by unicast to $P_{j,m+1}$ a message containing the new subgroup key $K'_{SGj}$, inverses of secrets assigned to other members of the subgroup, and some other keys (these are not relevant for our observation). This message is encrypted using the key $K_j^{(m+1)}$.

We do not describe the protocol in greater detail, since the problem is apparently visible – knowing the inverse of the secret assigned to other members (and particulary $a^{-1}_{j,m+1}$) allows any member (without loss of generality say $P_{j,1}$) to compute $K_j^{(m+1)}$ and decrypt the message sent to $P_{j,m+1}$. From this message, $P_{j,1}$ learns $a^{-1}_{j,1}$. Hence, when leaving the subgroup, $P_{j,1}$ can compute the subsequent key just like remaining members:

$$\begin{aligned} (K'_{SGj})^{a^{-1}_{j,1}} \bmod p &= (g^{a_{j,1}\cdot\ldots\cdot a_{j,m+1}\cdot s'_j})^{a^{-1}_{j,1}} \bmod p \\ &= g^{a_{j,2}\cdot\ldots\cdot a_{j,m+1}\cdot s'_j} \bmod p. \end{aligned}$$

That means the protocol cannot guarantee the forward secrecy of multicast data.

## 3 Conclusion

We described obvious insecurity of the multicast protocol proposed in [2]. The proposal lacks any decent security analysis. Moreover, the protocol cannot be easily fixed, since not distributing the inverses of secrets assigned to other members of the subgroup (which is the root cause of its insecurity) makes the leave operation more complicated, and the contribution of the proposal vanishes.

## Acknowledgments

This work was supported by VEGA 1/0266/09.

## References

[1] M. M. N. Rasslan, Y. H. Dakroury, and H. K. Aslan, "A new secure multicast key distribution protocol using combinatorial boolean approach", *International Journal of Network Security*, vol. 8, no. 1, pp. 75-89, 2009.

[2] I. Aly Saroit, S.F. El-Zoghdy, and M. Matar, "A scalable and distributed security protocol for multicast communications", *International Journal of Network Security*, vol. 12, no. 2, pp. 61–74, 2011.

[3] R. Srinivasan, V. Vaidehi, R. Rajaraman, S. Kanagaraj, R. Chidambaram Kalimuthu, and R. Dharmaraj, "Secure group key management scheme for multicast networks", *International Journal of Network Security*, vol. 11, no. 1, pp. 33-38, 2010.

[4] L. Wang and C. Wu, "Efficient key agreement for large and dynamic multicast groups", *International Journal of Network Security*, vol. 3, no. 1, pp. 8-17, 2006.

[5] Q. Zhang and K. L. Calvert, "A peer-based recovery scheme for group rekeying in secure multicast", *International Journal of Network Security*, vol. 6, no. 1, pp. 15-25, 2008.

**Martin Stanek** is an Associate Professor in the Department of Computer Science, Comenius University. He received his PhD. in computer science from Comenius University. His research interests include cryptography and information security.