# An insecure noninteractive group key establishment scheme

Chris J. Mitchell
Information Security Group, Royal Holloway, University of London
www.chrismitchell.net

19th September 2020

**Abstract**

A serious weakness in the recently proposed Chen-Hsu-Harn group authentication and group key establishment scheme is described. A simple attack against the group key establishment part of the scheme is given, which casts doubt on the viability of the scheme.

## 1 Introduction

A recent article by Cheng, Hsu and Harn proposed a combined (group) membership authentication and key establishment scheme. The scheme is claimed to be lightweight and hence suitable for wireless sensor networks (WSNs).

There is a very extensive literature on group key establishment schemes, many of which at least provide implicit authentication of the group members. The interested reader is referred to Boyd, Mathuria and Stebila [1]. It is far from clear whether, even it was secure (and it is not, as we describe below), the Chen-Hsu-Harn scheme offers any advantages over the state of the art, since the only comparison provided is with schemes using public key cryptography.

We describe a serious weakness in the Cheng-Hsu-Harn scheme [2]. The remainder of the paper is structured as follows. In Section 2 we briefly outline the operation of the scheme; this is followed in Section 3 by some brief observations on its functioning. The attack is described in Section 4, and concluding remarks are given in Section 5.

## 2 The scheme

The scheme involves a universally trusted *Membership Registration Centre* (*MRC*), which provides information to each of $n$ participating entities $\{U_1, U_2, \ldots, U_n\}$. This information enables any subset of the entities to authenticate each other 'as a group', and also to establish a shared secret key which is not available to participating entities not in the subset. The scheme uses arithmetic in GF($p$), the finite field of $p$ elements, for some prime $p > n$. No other requirements on $p$ are specified.

The scheme has five main stages, which we next briefly enumerate. The first stage is used to set up all the participants, and is only performed once. The remaining four steps are performed whenever a subset of entities wish to authenticate and establish a shared key. The reader is directed to the Cheng et al. paper [2] for the details — the notation used below is exactly as used in that paper.

**0. Token generation** This preliminary stage, performed once before active use of the scheme, involves the MRC generating and distributing a pair of 'shares' $(s_i(y),\ s_i(x))$ to each authorised participant $U_i$ $(1 \leq i \leq n)$, where $s_i(y)$ is a polynomial of degree $h-1$ over GF($p$) and $s_i(x)$ is a polynomial of degree $t-1$ over GF($p$), and where $h > 2t-2$.

**1. Pairwise key generation** In this first operational stage, the members of a 'group', i.e. a subset $\{U_{v_1}, U_{v_2}, \ldots, U_{v_m}\} \subseteq \{U_1, U_2, \ldots, U_n\}$, compute pairwise secret keys $k_{i,j}$ for each other using their shares. In fact, this step could be performed just once as part of the initialisation process, since the pairwise keys will always be the same.

**2. Group authentication** This involves the members of the group mutually authenticating each other using the pairwise secret keys $k_{i,j}$. After this step has completed each participant is confident that all members of the group agree on which entities are in the group.

**3. Group key establishment** This involves a further exchange amongst group members, as a result of which they agree on a shared secret key. In this exchange, information sent between group members is always encrypted using a pairwise shared secret key (as established in step 1). The group key is computed as the exclusive-or of values received from other group members.

**4. Group key authentication** This final stage, involving yet another exchange, is designed to give assurance that all members of the group agree on the shared secret key.

In this paper we describe an attack on the final two stages of the scheme, i.e. the group key establishment and group key authentication stages.

# 3 Some observations

Before describing the attack, we make some minor observations on the operation of the scheme.

- There is no direct link between the group authentication stage and the group key establishment stage, except for the set of identities of the participants in the 'group'.
- The nature of the encryption function $E$ used in group key establishment is not specified. We assume here that it is instantiated as authenticated encryption (to avoid attacks that might be possible if encrypted values could be manipulated).
- The scheme involves computing the bitwise-exclusive-or of values computed module $p$. We assume here that prior to applying the exclusive-or operation the values are converted from integers to bit strings.

# 4 An attack on group key establishment

## 4.1 Attack scenario, attack model and attack objective

We suppose that a set of $m$ ($m \leq n$) participants $\{U_{v_1}, U_{v_2}, \ldots, U_{v_m}\}$ have successfully completed the group authentication stage.

We further suppose that an (insider) adversary $U_{v_k}$ ($1 \leq k \leq m$) controls the broadcast channel with respect to 'victim' participant $U_{v_j}$ ($1 \leq j \leq m$, $j \neq k$), i.e. the adversary can (a) prevent messages sent by other legitimate participants from reaching $U_{v_j}$, and (b) send messages to $U_{v_j}$ on this channel that appear to have come from other legitimate participants. Since the protocol makes no assumptions about the trustworthiness of the communications channels, this assumption is legitimate. Indeed, if the broadcast channel was completely trustworthy, then much of the protocol would not be needed.

The objective of the adversary is to make the victim accept a key that is different to the key that is accepted by all other members of the set $\{U_{v_1}, U_{v_2}, \ldots, U_{v_m}\}$. This would appear to negate the purpose of the group key authentication stage, which is (presumably) all about enabling all members of the 'group' to verify that they share the same key.

## 4.2 Subverting group key establishment

The adversary $U_{v_k}$ first chooses a key $K^*$ which it wishes the victim $U_{v_j}$ to (wrongly) accept as the shared group key. The adversary $U_{v_k}$ allows all

messages sent by other participants to reach their destinations correctly. However, the adversary sends two different versions of its own message:

- it sends an encrypted version of the 'correct' value $q_{v_k}$ to all participants $U_{v_s}$ $(1 \leq s \leq m)$ except for the victim $U_{v_j}$;
- it sends an encrypted version of the value $q_{v_k} \oplus K \oplus K^*$ to the victim $U_{v_j}$, where $K$ is the 'correct' shared group key.

Note that the adversary will need to wait until it has received all the values $q_{v_i}$ $(i \neq k)$ before it can send the value to the victim, since it must compute the group key $K$ before sending the value.

As a result of the above steps, all participants except for the victim $U_{v_j}$ will share the 'correct' group key $K$. However, the victim will believe that the group key is $K^*$. We observe in passing that:

- the adversary knows $K$ and $K^*$;
- this part of the attack does *not* require the adversary to manipulate the broadcast channel.

### 4.3 Breaking group key authentication

We conclude the attack by showing how the adversary can manipulate the authentication process so that all participants believe the protocol has concluded successfully. The authentication process requires each participant to broadcast $H(K||L)$ where $H$ is a cryptographic hash function, $K$ is the group secret key that has just been established, and $L$ is the sum of values broadcast (in cleartext) at the beginning of the key establishment process.

To complete the attack the adversary needs to take control of the broadcast channel to and from the victim $U_{v_j}$. The victim will broadcast $H(K^*||L)$ — the adversary suppresses this and masquerades as the victim to broadcast $H(K||L)$. All other participants will broadcast $H(K||L)$; the adversary prevents these messages reaching the victim, and instead sends the victim 'fake' broadcasts of $H(K^*||L)$.

This completes the attack — all participants except the victim will believe that $K$ is shared by the group, and the victim will believe $K^*$ is shared by the group.

## 5 Concluding remarks

We have demonstrated a simple attack which completely negates the objectives of the protocol. This means that the protocol should not be used.

Fundamentally, the fact that the authors have not provided rigorous proofs of security for the scheme means that attacks such as that described here remain possible. It would have been more prudent to follow established wisdom and only publish a scheme of this type if a rigorous security proof had been established. Similar remarks apply to the all-too-often misconceived attempts to fix broken schemes, unless a proof of security can be devised for a revised scheme. Achieving this in an efficient way seems difficult for this scheme.

## References


[1] C. Boyd, A. Mathuria, and D. Stebila, *Protocols for authentication and key establishment*, 2nd ed., Information Security and Cryptography, Springer, 2020.

[2] Q. Cheng, C. Hsu, and L. Harn, *Lightweight noninteractive membership authentication and group key establishment for WSNs*, Mathematical Problems in Engineering **2020** (2020), no. 1452546.