DOI 10.12694/scpe.v20i1.1440
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 1, pp. 93–99. http://www.scpe.org © 2019 SCPE

ON THE SECURITY OF AUTHENTICATED GROUP KEY AGREEMENT PROTOCOLS

SUMAN BALA,[*] GAURAV SHARMA,[†] HIMANI BANSAL,[‡] AND TARUNPREET BHATIA[§]

Abstract. The group key agreement protocol enables to derive a shared session key for the remote members to communicate securely. Recently, several attempts are made to utilize group key agreement protocols for s ecure multicasting in Internet of Things. This paper contributes to identify the security vulnerabilities in the existing protocols, to avoid them in future constructions. The protocols presented by Gupta and Biswas have been found insecure to ephemeral secret key leakage (ESL) attack and also, malicious insiders can impersonate an honest participant. Additionally, the protocol presented by Tan is also ESL-insecure. We also present a fix to the Tan's protocol to make it secure.

Key words: Group key agreement, Authentication, Insider security, Mutual authentication

AMS subject classifications. 68M12

1. Introduction. The recent security concerns are prevailing when multiple devices over a wireless communication interact among them leaking sensitive information to a non-participating entity [20, 19]. A common concern in these real world applications is to establish a secure session among the interested remote participants. It is always challenging to derive a shared secret key which can prevail over the advantages of adversary. A group key agreement (GKA) protocol enables the participating members to establish such a symmetric session key, usually following an asymmetric procedure. This symmetric key is further used for encryption decryption purpose. Various real life applications of GKA includes distributed computations, video conferencing and multi-user games. A variety of key establishment approaches have been presented in literature based on the network characteristics, agreement strategy, communication rounds and contributiveness. The two major classes representing the protocols are either transportation of session key or agreement via participant's contribution. In the key transport protocols, the session key is derived by one of the powerful nodes and then transferred securely to all the members of the group. The common session key, derived by all the members following an interactive protocol, is known as key agreement protocols.

The hybridization of above two categories can originate another variety of protocols namely, balanced and imbalanced protocols. The balanced protocols are equally contributive protocols while in imbalanced, all the participants contribute but the major part of computations, such as signature verification, is performed by some powerful node. Other than preserving the basic attributes such as known key security and forward secrecy, *contributiveness* is an important aspect of a GKA protocol. By contributiveness, we mean that all the member's contributions are involved so that none of the member can predetermine the session key without incorporation of other members.

Following the Diffie and Hellman [8] work on two-party key exchange, there has been extensive efforts to convert their two-party key exchange protocol to multi-party key exchange protocol [6, 11, 21]. Among the most notable works, Joux's one round three-party key agreement protocol [13] is considered as a significant contribution for practical GKA protocol due to the functionality of pairing. Based on Joux's work [13], Barua et al. [1] have presented protocols of multi-party key agreement in two flavours *unauthenticated*- based on ternary trees and *authenticated*- from bilinear maps. Unfortunately their protocols are secure against passive adversaries only. As established by Bellare and Rogaway (Crypto'93) [2], to avoid man in the middle (MITM) attack, *authentication* is an essential security requirement for key exchange protocols.

The first contribution towards modeling provable security for authenticated key exchange (AKE), was commenced by Bresson et al. [3, 4, 5] but their protocol accounts $O(n)$ rounds, which is very expensive. Later in 2003, Katz and Yung [15] presented a scalable compiler to transform any unauthenticated GKA into an authenticated GKA with the additional cost of one round. Since the GKA study involves multiple participants, the consideration of malicious insider is a realistic scenario. Katz and Shin [14] firstly modeled the *insider*

[*] Université Libre de Bruxelles, Belgium (suman1005@gmail.com)

[†] Université Libre de Bruxelles, Belgium

[‡] Jaypee Institute of Information Technology, Noida, India

[§] Thapar University, India

security in GKA protocols. Gorantla et al. [9] studied that the compromise of long-term key of one participant should not enable the impersonation of any other participant. The improved security model which addresses the *forward secrecy* and *key compromise impersonation resilience* (KCIR) for GKA protocols to take into account authenticated key exchange (AKE) security and mutual authentication (MA) security. In 2011, their model was revisited and enhanced by Zhao et al. [30] where they addressed the ephemeral secret key leakage (ESL) attack. The extended model is the strongest model, as it takes into account both the leakage of secret key as well as the leakage of ephemeral key independently. However, Tseng et al. [23] argued about the insufficiency of UF-ACMA secure signature scheme and proposed a UF-ACM-ESL secure signature based on Schnorr [17].

In *identity based* setting, the first authenticated ID-based GKA protocol was formalized by Choi et al.[7] in 2004, but their scheme was found vulnerable to insider colluding attack [29]. In 2007, Shim [16] claimed that scheme in [7] is vulnerable to another insider colluding attack and improved the protocol. Unfortunately, none of these AGKA protocols could achieve the perfect forward secrecy. Perfect forward secrecy allows the compromise of long term secret keys of all participants maintaining all earlier shared secrets unrevealed. In 2011, Wu et al. [28] presented a provably secure ID-AGKE protocol from pairings, providing forward secrecy and security against the insider attacks. Later, Wu et al. [27] presented their first revocable ID-based AGKE (RID-AGKE) protocol, which is provably secure and can resist malicious participants as well. The main attraction of this protocol was efficient revocation of group members. However, the protocol takes three rounds but unable to identify malicious participants. In a subsequent improvement, Wu et al. [26] proposed an ID-based AGKE protocol, which can passively detect malicious participants and also proved its security against insider attacks. Although, the protocol was later found insecure against an insider colluding attack by [24]. Afterwards, a two round revocable ID-AGKE protocol was presented by Wu et al. [25] which can identify malicious participants. Another work on authenticated group key agreement protocol without pairing is presented by Sharma et al. [18]. Recently in 2017, Gupta and Biswas [10] presented an ECCbased AGKA protocol and claimed it computationally efficient. However, in this paper, we present security flaws in their construction and proved it insecure. All the above discussed protocols are balanced GKA protocols where all the participants contribute equally and derive a shared session key.

On the other hand, some imbalanced GKA protocols are also presented where one of the powerful node contribute more in the computational sense. A recent contribution to improve the computational efficiency by Islam et al. [12] is presented. This is an ECC-based ID-AGKA protocol for imbalanced mobile networks. The best feature of this protocol is pairing-free property. However, Tan [22] found the Islam et al. [12] construction insecure and improved it. We present an ESL attack on their improved work and attempt to fix it.

Rest of the paper is organized as follows: in Section 2, we introduce necessary definitions, corresponding hardness assumption for AGKA protocol and standard security model for AGKA. Section 3 and Section 4 describes the AGKA protocols and our attacks on their construction, followed by the conclusion Section 5.

2. Preliminaries and Definitions.

In this section, we introduce mathematical definitions, hardness assumptions, the notion of AGKA protocol and security model for it. If X is a set, then $y \overset{\$}{\leftarrow} X$ denotes the operation of choosing an element y of X according to the uniform random distribution on X.

2.1. Notations Used.

This section describes the preliminaries used for AGKA protocol. Table 2.1 shows the notations used throughout the paper.

2.2. Definitions and assumptions.

Definition 2.1 (Computational Diffie-Hellman Problem (CDHP)). Let $\mathbb{G}$ be an additive cyclic group (precisely an elliptic curve group) of order q with generator P. Let $CDH : \mathbb{G} \times \mathbb{G} \to \mathbb{G}$ be a map defined by

$$CDH(X,Y) = Z, \text{ where } X = aP, Y = bP \text{ and } Z = abP\,.$$

The *computational Diffie-Hellman problem* (CDHP) is to evaluate $CDH(X,Y)$ given $X, Y \overset{\$}{\leftarrow} \mathbb{G}$ without the knowledge of $a, b \in \mathbb{Z}_q^*$. (Note that obtaining $a \in \mathbb{Z}_q^*$, given $P, X \in \mathbb{G}$ is solving the elliptic curve discrete logarithm problem (ECDLP).)

Definition 2.2 (Computational Diffie-Hellman Assumption). Given a security parameter λ, let $\langle q, \mathbb{G}, P, X, Y, \rangle \leftarrow \mathfrak{G}(\lambda)$. The *computational Diffie-Hellman assumption* (CDHA) states that for any PPT algorithm $\mathcal{A}$ which

Table 2.1
Notations Used

Notation	**Description**
q	a large prime number
$\mathbb{F}_q$	finite field
$\mathbb{E}/\mathbb{F}_q$	elliptic curve defined on $\mathbb{F}_q$
$\mathbb{G}, \mathbb{G}_1$	cyclic additive group composed of the points on $\mathbb{E}/\mathbb{F}_q$
$\mathbb{G}_2$	cyclic multiplicative group composed of the points on $\mathbb{E}/\mathbb{F}_q$
P	generator of $\mathbb{G}$
$\hat{e}$	admissible bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$
$H_i(\cdot)(1 \leq i \leq n)$	secure one-way hash functions
k	security parameter
$param$	system parameters
U_i	A mobile node
U_n	The powerful node
ID_i	Identity of node $U_i(1 \leq i \leq n)$
n	number of participants
$\mathcal{C}$	Challenger, who is authoritative to respond adversary's query
$\mathcal{A}$	Adversary

attempts to solve CDHP, its *advantage*

$$\mathbf{Adv}_{\mathfrak{G}}(\mathcal{A}) := Prob[\mathcal{A}(q, \mathbb{G}, P, X, Y) = CDH(X, Y)]$$

is negligible in λ. We say that the (t, ϵ)-*CDH assumption* holds in group $\mathbb{G}$ if there is no algorithm which takes at most t running time and can solve CDHP with at least a non-negligible advantage ϵ.

2.3. AGKA Protocol. Let there are total n participants $U_1, U_2, \cdots, U_n$ and any subset with $(n \geq 2)$ can run the protocol (π). Each participant is provided a (public, private) key pair. In a protocol, we refer by *session* a running instance. Each participant is allowed to run multiple sessions concurrently. An i^{th} instance of the protocol is represented as Π_U^i, where U is the corresponding user or participant. We define two identities - the session identity sid_U^i which is the session dependent information computed by user U at it's i^{th} instance using the shared information in that session, and the partner identity pid_U^i which is a set of identities of the participants who are involved in generation of the session key with Π_U^i . We say an instance Π_U^i *accepts* when it computes a valid session key sk. We say instances Π_U^i and $\Pi_{U'}^j$,(for $\Pi_U^i \neq \Pi_{U'}^j$) are partnered iff (i) they have both accepted (ii) $sid_U^i = sid_{U'}^j$ (iii) $pid_U^i = pid_{U'}^j$. We further define the term *freshness.*

Definition 2.3 (Freshness). *An instance Π_U^i is referred to be fresh if it satisfies the following conditions:*

1. *If the instance Π_U^i is accepted, neither U_i nor any of its partnered instances, can query Reveal key oracle.*
2. *No participant is allowed to query Corrupt and Reveal Ephemeral Key simultaneously.*
3. *In a partnered instance between U_i and U_j, if an adversary $\mathcal{A}$ corrupts U_j, any message sent from U_j to U_i must actually come from U_j.*

2.4. Security Model for AGKA Protocol. We analyze the security of proposed protocol within the standard security frame of indistinguishability. For the purpose we define the following experiment between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$:

Setup: On input a security parameter 1^λ, the challenger $\mathcal{C}$ runs $\mathbf{KeyGen}(1^\lambda)$ to generate the public parameter *Params* and the system key pair (pk, msk) and gives the adversary $\mathcal{A}$ the public key pk. msk is the master secret of the system.

Queries: $\mathcal{A}$ can adaptively make the following queries:

- Execute(Π_U^i): Any time the adversary $\mathcal{A}$ can query for the complete transcripts of an honest execution among the users selected by himself.
- Send(Π_U^i, m): During the normal execution of the protocol, this query returns the reply generated by instance Π_U^i.
- Reveal Key (Π_U^i): When the oracle is accepted, this query outputs the group session key.

- Corrupt(U_i): This query models the reveal of long-term secret key. The participant is honest iff adversary $\mathcal{A}$ has not made any *Corrupt* query.
- Ephemeral Key Reveal(Π_U^i): This query models the reveal of ephemeral key of participant U_i for instance Π_U^i.
- Test(Π_U^i): This query can be made only once during the execution of protocol π. The challenger responds with a session key.

Challenge: During the Test query, the challenger randomly selects a bit $b \xleftarrow{\$} \{0,1\}$ and returns the real session key if $b = 0$ or a random value if $b = 1$.

Guess: $\mathcal{A}$ outputs its guess b' for b.

The adversary succeeds in breaking the security if $b' = b$. We denote this event by $Succ_{\mathcal{A}}$ and define $\mathcal{A}$'s advantage as $Adv_{\mathcal{A}}(1^k) \overset{\text{def}}{=} |2Pr[Succ_{\mathcal{A}}] - 1|$.

Definition 2.4 (AKE-Security). *Let $\mathcal{A}_{ake}$ be an adversary against AKE-security. It is allowed to make queries to the Execute, Send, RevealKey, Ephemeral Key Reveal, Corrupt oracles. It is allowed to make a single Test query to the instance Π_U^i at the end of the phase and given the challenge session key $sk_{ch,b}$ (depending on bit b). Finally $\mathcal{A}_{ake}$ outputs a bit b′ and wins the game if $(1) b = b'$ and (2) the instance Π_U^i is fresh till the end of the game. The advantage of $\mathcal{A}_{ake}$ is $Adv_{\mathcal{A}_{ake}} = |2Pr[Succ_{\mathcal{A}_{ake}}] - 1|$. The protocol is called AKE-secure if the adversary's advantage $Adv_{\mathcal{A}_{ake}}$ is negligible.* Below we recall the MA-security considering both types of adversaries, outsiders and insiders.

Definition 2.5 (MA-security with outsider KCIR). *Let $\mathcal{A}_{ma,out}$ be an outsider adversary against MA-security. Let pid_U^i be a set of identities of participant in the group with whom Π_U^i wishes to establish a session key and sid_U^i denotes a session id of an instance Π_U^i. $\mathcal{A}_{ma,out}$ is allowed to make queries to the Execute, Send, RevealKey, EphemeralKey Reveal, Corrupt oracles. $\mathcal{A}_{ma,out}$ breaks the MA-security with outsider KCIR notion if at some point there is an uncorrupted instance Π_U^i with the key sk_U^i and another party U' which is uncorrupted when Π_U^i accepts such that there are no other insiders in pid_U^i and the following conditions hold:*

- *there is no instance $\Pi_{U'}^{i'}$ with $(pid_{U'}^{i'}, sid_{U'}^{i'}) = (pid_U^i, sid_U^i)$ or,*
- *there is an instance $\Pi_{U'}^{i'}$ with $(pid_{U'}^{i'}, sid_{U'}^{i'}) = (pid_U^i, sid_U^i)$ which has accepted with $sk_{U'}^{i'} \neq sk_U^i$.*

Definition 2.6 (MA-security with insider KCIR). *Let $\mathcal{A}_{ma,in}$ be an insider adversary against MA-security. It is allowed to query Execute, Send, RevealKey, EmphemeralKey Reveal and Corrupt oracles. It breaks the MA-security with insider KCIR if at some point there is an uncorrupted instance Π_U^i which has accepted with the secret key sk_U^i and another party U' which is uncorrupted when Π_U^i accepts and*

- *there is no instance $\Pi_{U'}^{i'}$ with $(pid_{U'}^{i'}, sid_{U'}^{i'}) = (pid_U^i, sid_U^i)$ or,*
- *there is an instance $\Pi_{U'}^{i'}$ with $(pid_{U'}^{i'}, sid_{U'}^{i'}) = (pid_U^i, sid_U^i)$ which has accepted with $sk_{U'}^{i'} \neq sk_U^i$.*

3. Review of Tan's Identity-based Authenticated Group Key Agreement Protocol.

This section reviews the Tan's pairing-free ID-AGKA protocol for imbanced mobile networks. Tan's pairing-free ID-AGKA protocol consists of five phases namely, Setup phase, Key extraction phase, Key agreement phase, Remove phase and Join phase. The notions used throughout the paper are listed in Table......

Setup: For a given security parameter k, PKG does the following:

- Choose a k-bit prime q and generate a group $\mathbb{G}$ over the elliptic curve, where P is the generator of the group of prime order q.
- Choose the master key $x \in \mathbb{Z}_q^*$ and compute the system public key $P_{pub} = xP$.
- Choose cryptographic hash functions as follows:
  - $H_0 : \{0,1\}^* \times \mathbb{G} \times \cdots \times \mathbb{G} \times Z_q \to \{0,1\}^k$
  - $H_1 : \{0,1\}^* \times \mathbb{G} \to Z_q$
  - $H_2 : \{0,1\}^* \times \mathbb{G} \times \mathbb{G} \to Z_q$
  - $H_3 : \{0,1\}^* \to Z_q$
  - $H_4 : \{0,1\}^* \times \mathbb{G} \times \cdots \times \mathbb{G} \to Z_q$
- Publish the system parameters $\mathbb{G}, F_q, q, P, H_0(\cdot), H_1(\cdot), H_2(\cdot), H_3(\cdot), H_4(\cdot), P_{pub}$.

Key Extraction: Public key generator extracts the secret key of U_i with identifier ID_i as follows:

- Choose a number $r_i \in \mathbb{Z}_q^*$ and compute $R_i = r_iP$.
- Compute a Schnorr signature [17] about the identity ID_i as $x_i = r_i + xH_1(ID_i, R_i)$.

U_i checks if $x_iP = R_i + H_1(ID_i, R_i)P_{pub}$. If the equality holds, U_i takes (x_i, R_i) as the private long-term key.

Key Agreement: Each node $U_i(1 \leq i \leq n-1)$ randomly selects two numbers $a_i, b_i \in \mathbb{Z}_q^*$ and computes $T_i = a_ix_iP, V_i = b_iP, s_i = b_i + x_iH_2(ID_i, T_i, V_i) \mod q$. Next U_i sends the message $(ID_i, T_i, V_i, R_i, s_i)$ to the powerful node U_n.

Upon receiving the message $(ID_i, T_i, V_i, R_i, s_i)$, U_n executes the following operations:

- Compute $P_i = R_i + H_1(ID_i, R_i)P_{pub}$ and check if $s_iP = V_i + H_2(ID_i, T_i, V_i)P_i$ holds. If it holds, U_n authenticates U_i.
- Choose two random numbers $a_n, b_n \in \mathbb{Z}_q^*$ and compute the following:

$$T_n = H_3(a_n||x_n)P, V_n = b_nP, Z_i = H_3(a_n||x_n)T_i(1 \leq i \leq n-1),$$
$$s_n = b_n + x_nH_4(ID_n||Z_1||Z_2||\cdots||Z_{n-1}||V_n||T_n) \mod q$$

- Broadcast the message $(ID_n, V_n, R_n, s_n, Z_1, Z_2, \cdots, Z_{n-1})$ to the group U.
- Compute the session key $SK = H_0(ID, Z, T_n, s_n)$, where $ID = ID_1||ID_2||\cdots||ID_{n-1}||ID_n$, $Z = Z_1||Z_2||\cdots ...|_{n-1}$.

Each U_i computes $P_n = R_n + H_1(ID_n, R_n)P_{pub}, T = (a_ix_i)^1Z_i$, and checks if $s_nP = V_n + H_4(ID_n, Z, V_n, T)P_n \mod q$. If it is valid, U_i computes the group session key $SK = H_0(ID, Z, T, s_n)$.

3.1. Our Attack and Fix. Recall that, when the leakage of ephemeral secret is included in the security model, the leakage of these short term secrets should not allow the adversary to compute the session key. In Tan's protocol, the leakage of a_i and b_i will allow the adversary to find long term secret key from the signature. The adversary can compute x_i as $x_i = (H_2(ID_i, T_i, V_i))^{-1}(s_i - b_i)modq$.

The adversary computes $T = (a_ix_i)^{-1}Z_i$ where Z_i can be easily eavesdropped from the transcript. The session key can be computed as $SK = H_0(ID, Z, T, s_n)$, where $ID = ID_1||ID_2||\cdots||ID_{n-1}||ID_n$ and $Z = Z_1||Z_2||\cdots||Z_{n-1}$. To fix the above attack, one solution is to use a signature such that the leakage of private key can be avoided on the leakage of ephemeral secrets while other solution suggests to mask the ephemeral secret. The masking can be done by a simple substitution $\tilde{b}_i = H_5(b_i, x_i)$, where $H_5 : \mathbb{Z}_q^* \times \mathbb{Z}_q^* \rightarrow \mathbb{Z}_q^*$. Now, the leakage of ephemeral secrets a_i and b_i will not allow the adversary to compute x_i from the signature.

4. Gupta and Biswas ECC-based AGKA Protocol. In this section, we first present the AGKA protocol by Gupta and Biswas and then, we discuss about the security vulnerabilities in their proposal. The protocol is insecure against insider colluding attack and ephemeral key leakage attack. The algorithm steps are as follows:

Setup(1^λ) : On input security parameter 1^λ, this phase outputs the system parameters *Params* in the following steps:

- Chooses an elliptic curve group $\mathbb{G}_1$ of prime order q . Let P be a generator of group $\mathbb{G}$. Let $\hat{e}$ be an admissible bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, where $\mathbb{G}_2$ is a multiplicative group.
- Chooses cryptographic hash functions $H_0 : \{0,1\}^* \rightarrow \mathbb{G}_2$, $H_1 : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{Z}_q^*$.
- Finally publishes the system parameters $Params = \{\mathbb{G}_1, \mathbb{G}_2, q, P, H_0(\cdot), H_1(\cdot), \hat{e}\}$.

KeyGen$(params, ID_i)$: The phase performs the following for all the group members:

- Each party P_i publishes her public key $PU_i = s_iP$ and computes her private key as $PR_i = \dfrac{s_i}{s_i + QU_i}P$, where $QU_i = H_0(ID_i)$ and $s_i \in \mathbb{Z}_q^*$ is a randomly selected master key of each party P_i.

Key Agreement(x_i, pid) : In this phase, all the participating group members $P_1, P_2, \cdots, P_n$ have their indexes in cyclic form and also, all the members have already received their public/private key pairs. The steps for key agreement protocol are as follows:

Round 1:

- Each party $P_i(1 \leq i \leq n)$ chooses a random $r_i \in \mathbb{Z}_q^*$ and computes $\psi_i = r_iP$ and $h_i = H_1(\psi_i, r_i)$.
- Each Party computes the signature on h_i as $\sigma_i = h_iPR_i$ and broadcasts $\langle\psi_i, h_i^{-1}, \sigma_i\rangle$.
- On receiving these signatures $\langle\psi_j, h_j^{-1}, \sigma_j\rangle$, all participants verify as $\hat{e}(\sigma_j, h_j^{-1}Q_{ID_j}) = \hat{e}(PU_j, P)$, where $Q_{ID_j} = H_0(ID_j)P + PU_j$.

Round 2: On successful verification, each P_i computes $X_i = r_i(\psi_{i+1} - \psi_{i-1})$ and broadcast to all other participants.

Key Computation : Each party computes the shared group key as
$K_i = nr_i\psi_{i-1} + Y_i = (r_1r_2 + r_2r_3 + ... + r_nr_1)P$
where $Y_i = (n-1)X_i + (n-2)X_{i+1} + ... + X_{i-2}$.

4.1. Our Attacks. Here, we list some attacks on the above protocol and prove that the protocol is not secure. The attack points are as follows:

Key Generation Flaw: In this protocol, it is ambiguous to derive the public/private key pair by the participant itself. Usually, there are three cryptosystems in practice, Public Key Infrastructure (PKI), Identity based Cryptosystem (IBC) and Certificateless Cryptosystem (CL-PKC). In all the cryptosystems, the private key is partially or fully generated by the trusted third party. If the user can derive the public/private key pair by themselves, there will be no authentication because an adversary can do the same and hence, anyone becomes a valid member in any communication.

Insider Colluding Attack: The presented protocol is also vulnerable to insider colluding attack. Two insiders P_{i-1} and P_{i+1} can collude together to impersonate the participant P_i in any other group. The malicious participants eavesdrop the transcript $\langle \psi_i, h_i^{-1}, \sigma_i \rangle$ from the previous session and replay this in a new group. Further, note that the computation of X_i in **Round** 2 can be easily performed by P_{i+1} and P_{i-1}.

$$\begin{aligned} X_i &= r_i(\psi_{i+1} - \psi_{i-1}) \\ &= r_i\psi_{i+1} - r_i\psi_{i-1} \\ &= r_{i+1}\psi_i - r_{i-1}\psi_i \end{aligned}$$

The common group key can be computed as $K_i = nr_{i-1}\psi_i + Y_i$
where $Y_i = (n-1)X_i + (n-2)X_{i+1} + ... + X_{i-2}$.
Therefore, any two malicious insiders can impersonate a participant without his consensus and agree upon some session key. The adversary in our attack must be an active adversary which has the privilege to call *Send* oracle in standard security model.

Ephemeral Key Leakage Attack: Another drawback of the scheme is, the leakage of ephemeral key directly compromises the group session key. However, the authors claim in Theorem 7.9 [10], the session key resistance against the leakage of session specific temporary information but the given session key formula $K_i = nr_i\psi_{i-1} + Y_i$ is completely dependent on r_i.

5. Conclusion. In this paper, we analyze two AGKA protocols against the claimed security notions and we found them insecure. The Gupta and Biswas protocol is vulnerable to ESL attack as well as insider colluding attack while the Tan's AGKA protocol is ESL insecure. We also present a fix to the Tan's protocol.

REFERENCES

[1] R. Barua, R. Dutta, and P. Sarkar, *Extending joux's protocol to multi party key agreement*, in Indocrypt, vol. 2904, Springer, 2003, pp. 205–217.

[2] M. Bellare and P. Rogaway, *Entity authentication and key distribution.*, in Crypto, vol. 93, Springer, 1993, pp. 232–249.

[3] E. Bresson, O. Chevassut, and D. Pointcheval, *Provably authenticated group diffie-hellman key exchange-the dynamic case*, in Asiacrypt 2001, vol. 2248, Springer, 2001, pp. 290–309.

[4] ———, *Dynamic group diffie-hellman key exchange under standard assumptions*, in Advances in CryptologyEUROCRYPT 2002, Springer, 2002, pp. 321–336.

[5] E. Bresson, O. Chevassut, D. Pointcheval, and J.-J. Quisquater, *Provably authenticated group diffie-hellman key exchange*, in Proceedings of the 8th ACM conference on Computer and Communications Security, ACM, 2001, pp. 255–264.

[6] M. Burmester and Y. Desmedt, *A secure and efficient conference key distribution system*, in Workshop on the Theory and Application of of Cryptographic Techniques, Springer, 1994, pp. 275–286.

[7] K. Y. Choi, J. Y. Hwang, and D. H. Lee, *Efficient id-based group key agreement with bilinear maps*, in PKC 2004, Springer, 2004, pp. 130–144.

[8] W. Diffie and M. Hellman, *New directions in cryptography*, IEEE transactions on Information Theory, 22 (1976), pp. 644–654.

[9] M. C. Gorantla, C. Boyd, and J. M. G. Nieto, *Modeling key compromise impersonation attacks on group key exchange protocols*, in PKC 2009, Springer, 2009, pp. 105–123.

[10] D. S. Gupta and G. Biswas, *An ecc-based authenticated group key exchange protocol in ibe framework*, International Journal of Communication Systems.

[11] I. Ingemarsson, D. Tang, and C. Wong, *A conference key distribution system*, IEEE Transactions on Information theory, 28 (1982), pp. 714–720.

[12] S. H. Islam and G. Biswas, *A pairing-free identity-based authenticated group key agreement protocol for imbalanced mobile networks*, Annals of télécommunications-annales des telecommunications, 67 (2012), pp. 547–558.

[13] A. Joux, *A one round protocol for tripartite diffie–hellman*, in International algorithmic number theory symposium, Springer, 2000, pp. 385–393.

[14] J. Katz and J. S. Shin, *Modeling insider attacks on group key-exchange protocols*, in Proceedings of the 12th ACM conference on Computer and communications security, ACM, 2005, pp. 180–189.

[15] J. Katz and M. Yung, *Scalable protocols for authenticated group key exchange.*, in Crypto, vol. 3, Springer, 2003, pp. 110–125.

[16] S. Kyung-Ah, *Further analysis of id-based authenticated group key agreement protocol from bilinear maps*, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 90 (2007), pp. 295–298.

[17] C.-P. Schnorr, *Efficient identification and signatures for smart cards*, in Conference on the Theory and Application of Cryptology, Springer, 1989, pp. 239–252.

[18] G. Sharma, R. A. Sahu, V. Kuchta, O. Markowitch, and S. Bala, *Authenticated Group Key Agreement Protocol Without Pairing*, in International Conference on Information and Communications Security, Springer, 2017, pp. 606–618.

[19] K. Sharma and B. Gupta, *Attack in smartphone wi-fi access channel: State of the art, current issues, and challenges*, in Next-Generation Networks, Springer, 2018, pp. 555–561.

[20] ———, *Taxonomy of distributed denial of service (ddos) attacks and defense mechanisms in present era of smartphone devices*, International Journal of E-Services and Mobile Applications (IJESMA), 10 (2018), pp. 58–74.

[21] M. Steiner, G. Tsudik, and M. Waidner, *Key agreement in dynamic peer groups*, IEEE Transactions on Parallel and Distributed Systems, 11 (2000), pp. 769–780.

[22] Z. Tan, *An efficient pairing-free identity-based authenticated group key agreement protocol*, International Journal of Communication Systems, 28 (2015), pp. 534–545.

[23] Y.-M. Tseng, T.-T. Tsai, and S.-S. Huang, *Enhancement on strongly secure group key agreement*, Security and Communication Networks, 8 (2015), pp. 126–135.

[24] F. Wei, Y. Wei, and C. Ma, *Attack on an id-based authenticated group key exchange protocol with identifying malicious participants.*, IJ Network Security, 18 (2016), pp. 393–396.

[25] T.-Y. Wu, T.-T. Tsai, and Y.-M. Tseng, *A provably secure revocable id-based authenticated group key exchange protocol with identifying malicious participants*, The Scientific World Journal, 2014 (2014).

[26] T.-Y. Wu and Y.-M. Tseng, *Towards id-based authenticated group key exchange protocol with identifying malicious participants*, Informatica, 23 (2012), pp. 315–334.

[27] T.-Y. Wu, Y.-M. Tseng, and T.-T. Tsai, *A revocable id-based authenticated group key exchange protocol with resistant to malicious participants*, Computer Networks, 56 (2012), pp. 2994–3006.

[28] T.-Y. Wu, Y.-M. Tseng, and C.-W. Yu, *A secure id-based authenticated group key exchange protocol resistant to insider attacks.*, J. Inf. Sci. Eng., 27 (2011), pp. 915–932.

[29] F. Zhang and X. Chen, *Attack on an id-based authenticated group key agreement scheme from pkc 2004*, Information Processing Letters, 91 (2004), pp. 191–193.

[30] J. Zhao, D. Gu, and M. C. Gorantla, *Stronger security model of group key agreement*, in Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, ACM, 2011, pp. 435–440.

Edited by: Khaleel Ahmad
Received: Sep 13, 2018
Accepted: Feb 20, 2019