City Research Online

City St George's, University of London

Citation: Komninos, N., Tselikis, C. & Douligeris, C. (2013). SAnoVs: Secure Anonymous Voting Scheme for clustered ad hoc networks. Paper presented at the 18th IEEE Symposium on Computers and Communication (ISCC'13), 07 - 10 July 2013, Croatia.

This is the unspecified version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: https://openaccess.city.ac.uk/id/eprint/2499/

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to City Research Online policy.

City Research Online: http://openaccess.city.ac.uk/ publications@citystgeorges.ac.uk

SAnoVS: Secure Anonymous Voting Scheme for Clustered Ad Hoc Networks

Komninos N.
Department of Computer Science
University of Cyprus
P.O. Box 20537 1678 Nicosia, Cyprus
nkom@ieee.org

Tselikis C.
Electronics Systems & Applications Design Dept.
Hellenic Aerospace Industry S.A.
P.O. Box 23 32009, Schimatari, Greece
ctselikis@gmail.com

Douligeris C.
Department of Informatics
University of Piraeus
80 Karaoli & Dimitriou Str., Piraeus 185 34, Greece
cdoulig@unipi.gr

***Abstract*— In this paper we propose a secure anonymous voting scheme (SAnoVS) for re-clustering in the ad-hoc network. SAnoVS extends our previous work of degree-based clustering algorithms by achieving anonymity and confidentiality of the voting procedure applied to select new cluster heads. The security of SAnoVS is based on the difficulty of computing discrete logarithms over elliptic curves, the intractability of inverting a one-way hash function and the fact that only neighboring nodes contribute to the generation of a shared secret. Furthermore, we achieve anonymity since our scheme does not require any identification information as we make use of a polynomial equation system combined with pseudo-random coordinates. The security analysis of our scheme is demonstrated with several attacks scenarios.examined with several attack scenarios and experimental results.**

Keywords-anonymity, voting clustering, shared key

I. INTRODUCTION

The self-organization and self-protection of autonomous wireless ad hoc networks remains open field for novel solutions. We address the self-organization of ad hoc networks by allowing the nodes to make independent decisions and to vote for their cluster head (CH) via the use of distributed re-clustering algorithms. We address the protection of such clustered networks with the adoption of a conference key distribution system (CKDS) used to establish a shared symmetric key between the neighboring ad hoc nodes. The session key protects the ad hoc communications and especially the re-clustering procedure which is achieved with voting. The whole scheme is the Secure Anonymous Voting scheme (SAnoVS).

A. *Our contributions are:*

- *Autonomous decisions*: the ad hoc nodes use a weighted degree-based clustering criterion in order to choose a neighboring candidate to act as cluster head. The nodes vote for that candidate cluster head node.
- *Secure re-clustering*: In SAnoVS an anonymous secured voting procedure is used to re-cluster the ad hoc network. The maximum number of votes collected is taken into account for the selection of a new local CH. The nodes use SAnoVS to securely communicate their autonomous votes (opinions).
- *Secure communications*: SAnoVS encapsulates an anonymous CKDS scheme which establishes an ephemeral symmetric shared secret (session key) among the cluster members based on ECDLP and localization techniques. The votes sent to the CH node that initiated the voting procedure and the rest messages exchanged amongst the members can be symmetrically encrypted with this ephemeral session key.
- *Anonymous communication*: SAnoVS does not require any node identification information as it makes use of a polynomial equation system with Langrage interpolation and node positioning.
- *Security vulnerabilities:* SAnoVS overcomes the identified weaknesses of the previous CKDS as described in section II.

II. REVIEW OF RELATED WORKS

In this section we focus our review on previous CKDS. In [1] the CKDS concept was first introduced. In [2] a CKDS with user anonymity based on an algebraic approach was proposed with the use of one-way hash functions to hide the identities of the attendants. In [3] two improvements of [2] were proposed. In 2003 an ECDLP-based scheme was proposed in [4] (Yang et al. scheme). In this scheme the CK is randomly chosen by the chair person who then broadcasts to the attendants the values y_i that belong to a linear curve. In

2004 the authors of [5] (Lin et al. scheme) modified the Yang et al. scheme because it was vulnerable to the attack of easily solving a set of linear equations to acquire the session key. The authors of [5] proposed modification of the transmitted values y_i as $y_i' = h_i \oplus y_i$.

In [6] (Kim et al. scheme) anonymity was achieved by introducing the Lagrange polynomial interpolation by which means even the chair person calculates the shared CK and also the Lagrange coefficients c_i are broadcasted instead of the values y_i. However, there are several weaknesses regarding those previous works:

- Both schemes [4] and [5] do not really maintain the user anonymity since the values y_i or y_i' distributed to the attendants are directly linked to their identity.
- Trying to keep some kind of anonymity in [4] and [5] would lead to unnecessary computation costs for key recovery and key verification by all the attending users.
- The scheme in [6] assumes that the private keys x_i are distributed to the nodes through a secure channel, which is unsafe because increases the chances to solve the ECDLP.
- In the polynomial interpolation used in Kim et al. scheme if n is small (i.e., five or less attending nodes) then the Lagrange polynomial would have a very small degree $n-1$ and in conjunction with a poorly designed Elliptic Curve cryptosystem (ECC) it could be solved by an attacker.
- The Kim scheme [6] still depends on the identities of the attendants for the calculation of the hash values h_i.
- In the Kim scheme [6], if the private key x_i is found by solving the ECDLP then an attacker (attendant or not) by brute force attack could find the corresponding identity and break the anonymity of the system (knowledge of who owns a specific private key).

Considering the above weaknesses, it is essential to propose a new security anonymous scheme for re-clustering in ad hoc networks.

III. THE PROPOSED SCHEME

In order to protect the voting scheme described in Part B of this section, where an ad-hoc cluster changes its head upon node decisions, we propose the SAnoVS scheme. Our SAnoVS follows the same principles of a threshold secret sharing scheme (TSS) and consists of three efficient algorithms: the public parameter generation (PG), the dealer setup (DS) and the share combiner (SC), to distribute a share secret, which we refer to as conference key (CK).

Public Parameter Generation (PG): Initially, each cluster head publicly chooses an elliptic curve E over a finite field $GF(q)$ and a base point G of order p. Then, each node secretly chooses pseudo-random coordinates, $x_i, y_i \in [1, p-1]$ that define a point Z_i, and broadcasts the corresponding public key $Q_i = Z_i G$ to each node $U_i \in A$ (let $A = \{U_1, U_2, \dots, U_m\}_i$ denote the set of all m nodes in the ad-hoc network).

Dealer Setup (DS): In order the cluster head, U_c, to distribute the shared secret in cluster members B, it computes the pair-wise keys $k_{ci} = Z_c Q_i$ shared with each U_i. Then, U_c computes the hash value $h_i = H(k_{ci} \,||\, Z_c \,||\, Z_i \,||\, T) \,||\, m$ and constructs a polynomial with degree $n-1$ using n points $(h_i, H(h_i))$ by applying Langrage polynomial interpolation, similar to [9].

$$f(Z) = \sum_{k=1}^{n} H(h_k) L_k(Z) \; mod \; p$$

$$\text{where, } L_k(Z) = \frac{\prod_{j=1, j\neq k}^{n} ((Z) - h_j)}{\prod_{j=1, j\neq k}^{n} (h_k - h_j)} = c_{n-1} Z^{n-1} + c_{n-2} Z^{n-2} + \cdots + c_1 x + c_0 \; mod \; p \quad (1)$$

Hence, the shared secret is the constant value of (1), $CK = c_0$. Next, U_c computes the check value of the share secret and add a timestamp T, as $V = H(CK \,||\, Z_c \,||\, T)$ before U_c broadcasts the message:

$$M = (Z_c, V, T, c_{n-1}, c_{n-2}, \dots, c_1) \quad (2)$$

In order to prevent small degree of polynomial which translates to small number of neighboring nodes, U_c generates additional pseudo-random coordinate pairs $(h_i, H(h_i))$ to increase the number of points available.

Share Combiner (SC): In this stage, each U_i in the cluster receives the message M and performs the share combiner recovery procedure, where only $U_i \in B$ can recover the correct CK after Step1 to Step4.

Step1. First, U_i verifies the expiration of the received timestamp, T and if it is invalid, U_i terminates the recovery process.

Step2. Second, U_i computes the shared pseudo-random coordinates with U_c, as $k_{ci} = Z_i Q_c$.

Step3. Third, U_i computes $h_i = H(k_{ci} \,||\, Z_c \,||\, Z_i \,||\, T) \,||\, m$ and solves CK from the following equality:

$$H(h_i) = c_{n-1}(h_i)^{n-1} + c_{n-2}(h_i)^{n-2} + \cdots + c_1 x + c_0 \bmod p$$
$$\xrightarrow{yields} CK = c_0 = H(h_i) - c_{n-1}(h_i)^{n-1} - c_{n-2}(h_i)^{n-2} - \cdots - c_1 x \bmod p \quad (3)$$

Step4. Finally, U_i checks the validity of CK by verifying

$$H(CK \,||\, Z_c \,||\, T) = V. \quad (4)$$

Only a valid member of the cluster $U_i \in B$ can recover the valid shared secret, CK, from the above equation with the use of session key k_{ci} shared with U_c. In SAnoVS, we construct our polynomial without using identities and we compute the shared secret, CK, by using polynomial equation system in SC stage. Therefore, SAnoVS does not require any user identification information or unecessary computation costs for

the attending members of the cluster. During the SAnoVS voting procedure the attending members cannot exchange their encrypted votes with the long-term session key k_{ci} to choose the next cluster head but only with the ephemeral key CK to avoid chosen plaintext attacks.

A. Local Candidate Selection

The highest degree algorithm [7] is a well-known ad hoc clustering algorithm in which as local CH is selected the node with the maximum connectivity degree, i.e., the node having the maximum number of uncovered in-range neighbors (periodic broadcast *hello* messages are used by the ad hoc nodes for one-hop neighbor detection). We adopt here for candidate selection a weighted clustering variable V_i which is a simplified variation of the clustering criterion defined in [8]. In more detail, we assume that the energy e_i along with the connectivity degree d_i of each node i are included in the *hello* broadcasts. Then, each node participating in the re-clustering procedure calculates the value of V_i for each neighbor i, including itself, by using the coefficient α (weighs degree and energy):

$$V_i = \alpha \cdot \frac{d_i}{d_{max}} + (1-\alpha) \cdot \frac{e_i}{e_{max}} \quad (5)$$

The neighbor with the maximum V_i constitutes the CH candidate node, opinion that the participating node will communicate during the SAnoVS voting procedure.

B. New Cluster Head Selection with Voting

The re-clustering procedure (selection of a new set of cluster heads to maintain a connected structure) is initiated in ad hoc networks given that some criteria are fulfilled. For example, in the LCC algorithm [10] re-clustering is initiated when two cluster heads come in range and in [11] lower overhead than LCC is demonstrated if the cluster head change is deferred for a period of time which depends on the speed of the two moving cluster heads that meet. We assume here that re-clustering is initiated by the current local CH when its energy drops below a specific threshold value [12]. In that case the following actions are taken place:

Step1: SAnoVS is taking place.

Step2: CH starts a *Voting_period* timer.

Step3: CH broadcasts a *Voting_initiation* message to his (*n* known) cluster members along with the $n-1$ Lagrange coefficients.

Step4: On reception of the *Voting_initiation* message the cluster members:

a) Recover the session key CK.

b) Apply our candidate selection procedure described in *A*, this section, to identify their vote (the resource-less current CH is excluded from the candidates).

c) Unicast to the current CH a *Voting_response* message including their pseudo-random coordinates and their candidate vote encrypted with the CK recovered from *a)*.

Step5: On reception of the *Voting_response(s)* the current CH checks for double votes received from exactly the same pseudo-random coordinates and if no duplicate exists stores the encrypted votes in a *Voting _table* otherwise drops the duplicate votes.

Step6: When the *Voting_period* timer expires, the CH decrypts the secured votes using the CK and stores the collected votes per each candidate in the *Voting _table*. New CH is the node with the maximum number of collected votes. The current CH broadcasts a *CH_Announcement* message with the ID of the elected new CH.

Step7: The nodes that hear the *CH_Announcement* message unicast a *Registration* message including their ID to affiliate with the new CH.

Step8: The new CH collects the memberships and unicasts a *Confirmation* message to each member (leadership is now ceded).

IV. Security Analysis

Our scheme follows well-defined cryptographic assumptions: the intractability of computing the elliptic curve discrete logarithm problem (ECDLP), the hardness of inverting a one-way function and the pseudo-randomness of the coordinates. If these assumptions can be solved easily, then SAnoVS cannot provide user anonymity and data privacy. Considering that, each cluster node $U_i \in B$ dynamically generates an elliptic curve key pair, whose secret key $Z_i \in [1, p-1]$ i.e., $x_i, y_i \in [1, p-1]$, is already known to the cluster nodes and public key Q_i is broadcasted to them. Therefore, this section presents several attack scenarios to demonstrate the security of the proposed scheme.

Attack scenario 1: Assume an attacker captures Q_i and tries to find the secret key Z_i. In order to find the pseudo-random coordinates Z_i, the attacker either need to solve ECDLP or brute force the [1, p-1] space.

Attack scenario 2: Assume an attacker collects the message $M = (Z_c, V, T, c_{n-1}, c_{n-2}, \dots, c_1)$ in the Dealer Setup phase and then tries to find the identity of the cluster nodes. If an attacker knows the attending cluster node secret key Z_i, they can obtain the participants' identity from M. However, computing Z_i from the public value is equivalent to solving the ECDLP.

Attack scenario 3: Assume a cluster node $U_i \in B$ tries to find the identity of another neighboring node. The cluster nodes $U_i \in B$ can easily reconstruct the share secret CK. However, it is infeasible to find the identity of another neighboring node since the node identities are not included at any stage of the proposed scheme.

Attack scenario 4: Assume an attacker that does not belong to the cluster tries to reveal the common share secret CK from the message M in DS phase. The attacker first computes the hash value $h_i = H(k_{ci} \,||\, Z_c \,||\, Z_i \,||\, T) \,||\, m$, then tries to recover the share CK based on the knowledge of the message M. However, non-cluster node has not the ability to obtain h_i,

because the difficulty involved in generating the coordinates Z_i is based on the ECDLP.

Attack scenario 5: An attacker tries to replay an intercepted message $M = (Z_c, V, T, c_{n-1}, c_{n-2}, \dots, c_1)$ to impersonate the cluster head U_c to hold the voting procedure. The attacker shoud set a new acceptable timestamp T, so that the cluster nodes can verify the validity of T in DS phase. Then, the cluster nodes compute k_{ci} and h_i to solve the CK and ckeck the validity of CK by verifying $H(CK \,||\, Z_c \,||\, T) = V$. However, the attacker can not forge a valid CK without knowing Z_c from Q_c. To obtain Z_c from Q_c is equivalent to solving the ECDLP. The cluster nodes can verify the validity of V at SC recovery stage. Therefore, an attacker cannot obtain any secret by replaying an intercepted message of equation (2), i.e., $M = (Z_c, V, T, c_{n-1}, c_{n-2}, \dots, c_1)$.

V. CONCLUSION

When clusters need to select their head node in ad-hoc networks it is desirable all cluster members to participate in this procedure. The most popular selection procedure is through voting and vote privacy can be achieved through encryption with the use of a share secret key. Therefore, in the context of secure clustered ad hoc networks we have proposed improvements on calculating and distributing a shared secret key without revealing the identities of the participating nodes. Moreover, linear threshold schemes with elliptic curve cryptographic techniques are applied to distribute such shares. In particular, we have used pseudo-random coordinates of the participating nodes during the public key generation stage; during the set up of the ephemeral shared secret; and during voting of the new cluster heads. Our scheme has been evaluated with attack scenarios and proved that we overcome the vulnerabilities of the previously proposed schemes.

REFERENCES

[1] I. Ingemarsson, D.T. Tang, C.K. Wong, “A conference key distribution system”, IEEE Transactions on Information Theory IT-28 (1982), pp: 714–720.

[2] T.C. Wu, “Conference key distribution system with user anonymity based on algebraic approach”, IEE Proceedings. Computer Digital Technology 144 (2) (1997), pp: 145–148.

[3] Y.M. Tseng, J.K. Jan, “Anonymous conference key distribution systems based on discrete logarithm problem”, Computer Communications 22 (1999), pp: 749–754.

[4] C.C. Yang, T.Y. Chang, M.S. Hwang, “A new anonymous conference key distribution system based on the elliptic curve discrete logarithm problem”, Computer Standards and Interfaces.

[5] C.H. Lin, C.Y. Lee, W. Lee, “Comments on the Yang-Chang- Hwang anonymous conference key distribution system”, Computer Standards and Interfaces 26 (2004), pp: 171–174.

[6] W. H. Kim, E.K. Ryu, J.Y. Im, K.Y. Yoo, “New conference key agreement protocol with user anonymity”, Computer Standards & Interfaces vol. 27, 2005, pp: 185–190.

[7] M. Gerla and J. T.-C. Tsai. “Multicluster, mobile multimedia radio network”, Wireless Networks, 1995, 1:255–265.

[8] C. Tselikis, S. Mitrpoulos, C. Douligeris, N. Komninos, “Degree-based Clustering Algorithms for Wireless Ad Hoc Networks Under Attack", IEEE Communications Letters, Volume 16, Number 5, 2012, pp. 619-621.

[9] N. Komninos, C. Douligeris, “LIDF: Layered intrusion detection framework for ad-hoc networks”, Journal of Ad Hoc Networks, Volume 7, Issue 1, January, 2009, pp: 171-182.

[10] C.-C. Chiang, H.-K. Wu, W. Liu, Mario Gerla., “Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel,” Proceedings of IEEE SICON’97, October 1997.

[11] Deosarkar and R.P.Yadav, “A Low Control Overhead Cluster Maintenance Scheme for Mobile Ad hoc NETworks (MANETs)”, ACEEE International Journal on Network Security 1, 1 (2010) 5.

[12] Garth V. Crosby, Niki Pissinou, James Gadze, “A Framework for Trust-based Cluster Head Election in Wireless Sensor Networks”, DSSNS '06: Proceedings of the Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems (2006), pp. 13-22.