Group Key Based Session Key Establishment Protocol for a Vehicle Diagnostic

Sarang Wi(✉), Kiwoon Moon, Boohyung Lee, and Jong-Hyouk Lee

Protocol Engineering Laboratory, Sangmyung University,
Cheonan, Republic of Korea
{sarang,kiwoon,boohyung,jonghyouk}@pel.smuc.ac.kr
http://pel.smuc.ac.kr

Abstract. In this paper, a group key based session key establishment protocol is introduced for a remote vehicle diagnostic. The proposed scheme aims at providing secure authentication and session key establishment between a vehicle manufacturer server and a group of in-vehicle electronic control unit based on a key graph.

Keywords: Key graph · Group key · Authentication · Session key

1 Introduction

As wireless communication and networking technologies continue to develop, vehicular communication technologies are emerging. Connected cars are expected to be in our daily life soon. Various services like traffic information notification, location-based services, vehicle remote diagnostic will be available [1].

A connected car is considered as a computing device that moves along roads. BI Intelligence expects that the connected car occupied 75% of the world's automotive production in 2020 [2]. This computing device is not only providing web browsing, video streaming, etc. but also used as a vehicle that carries people inside. As like personal computers and smartphones, the connected car is connected to other cars and also to the Internet so that it can be a target by attackers. For instance, an attacker can obtain vehicle state information and use this maliciously control the vehicle speed and breaking system [3].

As a preliminary work, a session key establishment protocol for vehicle diagnostic has been investigated that was based on symmetric key cryptosystem [4]. It has some practical issues, e.g., the number of symmetric key increases as the number of the Electronic Control Units (ECUs) in a car increases [4]. In other words, in terms of key management, the preliminary work is inefficient. In order to address this issue, in this paper, we are focused on developing a secure session key establishment protocol for a vehicle diagnostic based on group key graphs.

The rest of the paper is organized as follows. In Sect. 2, we present some related works. In Sect. 3, we present the proposed scheme. Section 4 concludes this paper.

J.-H. Lee and S. Pack (Eds.): QShine 2016, LNICST 199, pp. 62–71, 2017.
DOI: 10.1007/978-3-319-60717-7_7

2 Background

2.1 In-Vehicle Communication

The remote diagnostic is a service that allows a vehicle manufacturer monitors a vehicle's status through ECUs in the vehicle. The obtained information can be used for vehicle diagnosis. For example we check tire, engine, turbo charger, etc., through the information [5,6]. The ECUs make up a larger percentage of in-vehicle electronic unit and the percentage will be increased countinously [7].

A in-vehicle network is mainly implemented as a Controller Area Network (CAN), which provides the most transmission speed to up 1 Mbit/s [8]. To overcome the inefficiency of CAN that is low-speed and data transmission mode, Flexray has been introduced [9]. Recently the use of Ethernet in a in-vehicle network has been considered [10].

2.2 Group Key

There is a case for transmitting secure data unto the only member of a group. If each user uses a pair of symmetric keys to encrypt the data, it would be inefficient in terms of network bandwidth, computation, key management cost. To overcome this limitation, a group key has been introduced that all member of a group shares a same key. The important part of a group key use is a group dynamic, which means that the member of the group can be changed. The requirements for group keying are thus as follows [11].

- Forward secrecy: When a member leaves from the group, the member who knows the old group key should not be able to know a new group key.
- Backward secrecy: when a member join to the group, the member who knows the current group key should not be able to know the old group key.

We use key graphs as a group key management model. We describe the idea of a secure group as (E, K, R) where E is a set of users, K is a set of keys, and R denotes $R \subset E * K$, which is a user-key relation. A key is held by each ECU in E.

We need a secure server to manage the group keys. The server should distribute safely the key unto a member of the ECU group and maintain the relation R between the ECU and the key of a group. Each ECU of the group has a set of keys: ECU's individual key k_{e_u}, a sub-group key, and a group key. The ECU's individual key is shared only with the key management server. Let Z is a group name. For Z, the group key is K_{Gr_z}, which is used to send a message securely to other ECU belonging to Z.

In this paper, we explain special classes of key graphs (i.e., Tree) to explain the key management. This key graph has two types: *e-nodes* representing ECUs and *k-node* representing keys. Each *e-node* has an outgoing edge but no incoming edge. Each *k-node* has an incoming edge. Among *k-node* top *k-node* is called as a root node. And the root node is single. This key graph has two parameters *height* and *degree*. *height* is the distance between the root node and uttermost

with root node and end node. *degree* is the maximum number of incoming edges in the tree.

Relation of key graph G and secure group (E, K, R) is as follows:

1. E and the set of *e-node* is an one to one correspondence in G.
2. K and the set of *k-node* is an one to one correspondence in G.
3. R constitutes (e, k). G shows a directed path between *e-node* that corresponds e and *k-node* that corresponds to k.

For the key management, suppose that there exists nine ECUs and the ECUs are divided into three subgroups. The subgroups are $\{e_1,\ e_2,\ e_3\}$, $\{e_4,\ e_5,\ e_6\}$, and $\{e_7,\ e_8,\ e_9\}$. Each ECU has three keys: individual key, entire group key, and subgroup key. The tree key graphs G in Fig. 1 specifies the following a secure group.

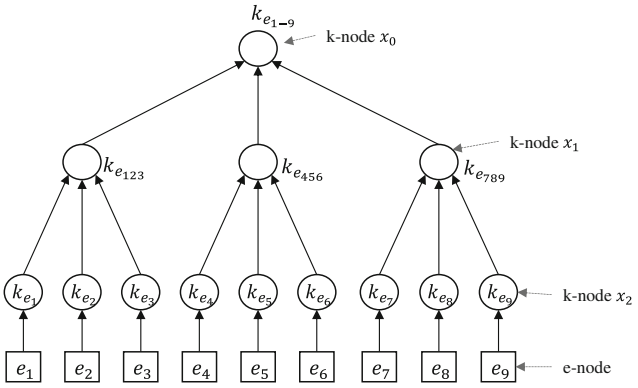

Fig. 1. Tree key graph

$$
\begin{aligned}
E = {} & \{e_1,\ e_2,\ e_3,\ e_4,\ e_5,\ e_6,\ e_7,\ e_8,\ e_9\} \\
K = {} & \{k_{e_1},\ k_{e_2},\ k_{e_3},\ k_{e_4},\ k_{e_5},\ k_{e_6},\ k_{e_7},\ k_{e_8},\ k_{e_9},\ k_{e_{123}},\ k_{e_{456}},\ k_{e_{789}},\ k_{e_{1-9}}\} \\
R = {} & \{(e_1, k_{e_1}),\ (e_1, k_{e_{123}}),\ (e_1, k_{e_{1-9}}), \\
& (e_2, k_{e_2}),\ (e_2, k_{e_{123}}),\ (e_2, k_{e_{1-9}}), \\
& (e_3, k_{e_1}),\ (e_3, k_{e_{123}}),\ (e_3, k_{e_{1-9}}), \\
& (e_4, k_{e_1}),\ (e_4, k_{e_{456}}),\ (e_4, k_{e_{1-9}}), \\
& (e_5, k_{e_1}),\ (e_5, k_{e_{456}}),\ (e_5, k_{e_{1-9}}), \\
& (e_6, k_{e_1}),\ (e_6, k_{e_{456}}),\ (e_6, k_{e_{1-9}}), \\
& (e_7, k_{e_1}),\ (e_7, k_{e_{789}}),\ (e_7, k_{e_{1-9}}), \\
& (e_8, k_{e_1}),\ (e_8, k_{e_{789}}),\ (e_8, k_{e_{1-9}}), \\
& (e_9, k_{e_1}),\ (e_9, k_{e_{789}}),\ (e_9, k_{e_{1-9}})\}
\end{aligned}
$$

Here, $k_{e_{1-9}}$ is a group key. The following two functions are defined for the secure group (E, K, R):

$$
\begin{aligned}
& keyset(e) = \{k \mid (e, k) \in R\} \\
& ECUset(k) = \{u \mid (e, k) \in R\}
\end{aligned}
$$

$keyset(e)$ is a set of keys that has ECU e in E. $ECUset(k)$ is a set of ECUs that has key k in K. For example, applying this function in Fig. 1, we have $keyset(e_3) = \{k_{e_3}, k_{e_{123}}, k_{e_{1-9}}\}$ and $ECUset(k_{e_{456}}) = \{e_4, e_5, e_6\}$.

All details about the key management using tree key graphs are available in [12], which have been also adopted in this paper.

3 Proposed Scheme

In this section, we present the proposed scheme designed for establishing a secure session key between a vehicle manufacturer server and a vehicle ECU based on a group key. The session key is then used for instance to encrypt and decrypt data communications between the server and ECU for a vehicle diagnostic.

3.1 Notation and Assumption

Before explaining used notations and assumptions, we explain main agents of communication. The main agents is server of vehicle manufacturer S, in-vehicle gateway Gw, vehicle ECU E_i. When a car is out of the shop, it establishes a session key through authentication between entities. Table 1 shows the notations used for the proposed scheme.

Table 1. Notation

Notations	Definition
S	vehicle manufacturer's server
Gw	in-vehicle gateway
E_i	vehicle ECU
Gr_z	vehicle ECU group z
ID_S	ID of S
ID_{Gw}	ID of Gw
ID_{E_i}	ID of E_i
ID_{Gr_z}	ID of Gr_z
J	secret key of Gw
K	shared key between S and Gw
Q	secret key of GW; used for creating a group key
K_{E_i}	secret key between Gw and E_i
K_{Gr_z}	group key of ECU group z
SK_{S-Gr_z}	session key between S and Gr_z
$E_{key}[]$	symmetric encryption
$D_{key}[]$	symmetric decryption
R	server's nonce
$h(.)$	one-way cryptographic hash function

The assumptions are as followings.

1. When the vehicle is shipped, S is authenticated with Gw. A secure channel between S and Gw is established.
2. When the vehicle is shipped, Gw have a three secure key J, K, Q.
3. When the vehicle is shipped, Gw and E_i share a secure key K_{E_i}.

3.2 Operation Process

Vehicle Registration for Server. When a vehicle is produced, Gw is registered with S. In this time, S sends ID_S unto Gw securely. Gw receives ID_S from S. And Gw computes $C_{ig} = E_J[ID_S||ID_{Gw}]$. Only Gw creates C_{ig}. Here, C_{ig} is used for mutual authentication between S and Gw. Gw sends C_{ig}, K unto S through a secure channel. Here, J, K are secure keys in Gw for a long time.

Authentication Between Vehicle and Serve. For communicate between S and E_i, Gw (i.e., key management server in a vehicle) makes a group key of each ECU group and sends a message including the group key that encrypted by each E_i's individual key. E_i decrypts the message using the own individual key. It is possible that it does encrypted communication between Gw and E_i. In Fig. 2, shows the session key establishment process between vehicle and S.

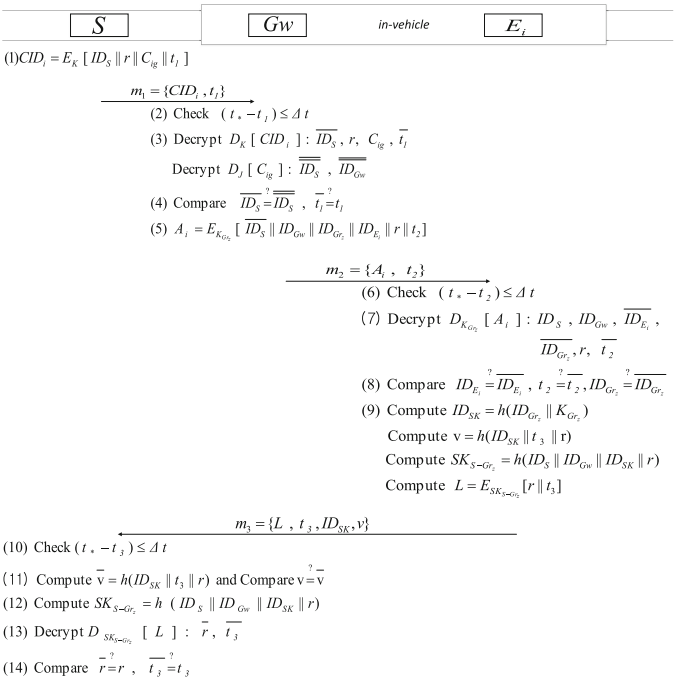

Fig. 2. Session key establishment between vehicle and S

1. S computes $CID_i = E_k[ID_S||r||C_{ig}||t_1]$. ID_S is server's ID. r is a random number. C_{ig} is received from Gw when the vehicle is manufactured. t_1 is the current time of S. Thereafter, S sends a message $m_1 = \{CID_i, t_1\}$ unto Gw. CID_i is used for authentication between S and Gw.
2. Upon receiving S's message, Gw performs the following steps. Check, if $(t_* - t_1) \leq \triangle t$, if yes, then Gw performs the next process. Otherwise, it rejects the request and aborts any further process. Here, $\triangle t$ is the time interval for the transmission delay and t_* is the current time of Gw.
3. Gw decrypts the message CID_i, using key K (i.e., $D_K[CID_i]$) and obtains $\overline{ID_S}$, r, C_{ig} and $\overline{t}_1$. Similarly, it decrypts the message C_{ig} using key J (i.e., $D_J[C_{ig}]$) and obtains $\overline{\overline{ID_S}}$ and $\overline{\overline{ID_{Gw}}}$.
4. Gw compares $\overline{ID_S} = \overline{\overline{ID_S}}$ and $t_1 = \overline{t_1}$, if yes, then Gw continues with the next steps; otherwise it aborts the request.
5. Gw computes $A_i = E_{K_{Gr_z}}[\overline{ID_S}||ID_{Gw}||ID_{Gr_z}||ID_{E_i}||r||t_2]$. Here, t_2 is the current time of Gw. Thereafter, Gw sends the message $m_2 = \{A_i, t_2\}$ unto E_i. A_i is used for authentication between Gw and E_i. This process delivers S's ID_S, r to create the session key of E_i.
6. Upon receiving Gw's message, the E_i performs the following steps. Check, if $(t_* - t_2) \leq \triangle t$, if yes, then E_i performs the next process. Otherwise, it rejects the request and aborts any further process. Here, t_* is the current time of E_i.
7. E_i decrypts the message A_i using group key K_{Gr_z} (i.e., $D_{K_{Gr_z}}[A_i]$), and obtains ID_s, ID_{Gw}, $\overline{ID_{E_i}}$, $\overline{ID_{Gr_z}}$, r, $\overline{t_2}$.
8. E_i compares $ID_{E_i} = \overline{ID_{E_i}}$, $t_2 = \overline{t_2}$, $ID_{Gr_z} = \overline{ID_{Gr_z}}$, if yes, then E_i continues with the next steps; otherwise it aborts the request.
9. E_i computes $ID_{SK} = h(ID_{Gr_z}||K_{Gr_z})$, $v = h(ID_{SK}||t_3||r)$, session key $SK_{S-Gr_z} = h(ID_S||ID_{Gw}||ID_{SK}||r)$, and $L = E_{SK_{S-Gr_z}}[r||t_3]$. Here, t_3 is the current time of E_i. After that, E_i sends the message $m_3 = \{L, ID_{SK}, v, t_3\}$ unto S.
10. Upon receiving the E_i's message, S validates the time as follows. Check, if $(t_* - t_3) \leq \triangle t$, if yes, then S performs the next process. Otherwise, S rejects the request and aborts any further process. Here, t_* is the current time of S.
11. S computes $\overline{v} = h(ID_{SK}||t_3||r)$, and compares $v = \overline{v}$, if yes, then S continues with the next steps; otherwise it aborts the request.
12. S computes session key $SK_{S-Gr_z} = h(ID_S||ID_{Gw}||ID_{SK}||r)$.
13. S decrypts the message L using key SK_{S-Gr_z} (i.e., $D_{SK_{S-Gr_z}}[L]$), and obtains $\overline{r}$, $\overline{t_3}$.
14. S compares $r = \overline{r}$, $t_3 = \overline{t_3}$, if yes, then a secure session key is established; otherwise not.

When the session key establishment is completed, E_i sends a message including the session key to other ECUs in the same group so that other ECUs will have the session key for secure communications with S.

3.3 Key Management Among the ECU

In this section, we present the key management for the ECUs based on a group key. We concentrate upon the group key when joining and leaving of the group. The group key cryptosystem should create a new key (i.e., new individual key, new subgroup key, new group key) for a group for joining and leaving events. Here, the creating new key should not analogize the whole out of an old key and should send that only the member within a group knows the key. Joining and leaving of proposed environment happened on two occasions. The first occurs registration that is all ECU into the server when the vehicle is shipped. The second occurs that the ECU's something the matter. When the second occasions, the ECU replaces the new ECU in a garage.

Joining. An ECU e which want to join the secure group sends requesting message to join to the key distribution server. This key server manages group and has access management authority. When is received the requesting message, the server begins an exchange for authentication ECU e. If join request is approved, as a result of the authentication the server and ECU e has the session key k_{e_u}. The server creates a new *e-node* of ECU e and new *k-node* of e's individual key. The server looks for joining point which we call parent node to attach newly created *k-node* in tree key graphs, and attaches this *k-node*. After the server creates new group key. To prevent a join of new member e in the past communication access, the key must be changed from joining point to root node. These keys must be delivered safely unto joining member and existing members. So the key management server encrypts these keys by previous group key for existing members or individual key for joining member.

If the server authorizes the ECU e and distribute the key k_{e_u} unto e, that thing as follows. The server finds a joining point and attach k_{e_u}. At the time x_j denotes the joining point, x_0 the root, and when $i = 1, \ldots, j$, x_{i-1} the parent of x_i. K_{j+1} denotes k_{e_u} and $K_0, \ldots, K_j$ the old keys of $x_0, \ldots, x_j$. The server generates new keys $K'_0, \ldots, K'_j$. The server sends $K'_0, \ldots, K'_j$ unto each ECU. When expressed as a function, it is $ECUset(K_0) : \{K'_0\}_{K_0}, \ldots, \{K'_j\}_{K_j}$. And it sends $\{K'_0, \ldots, K'_j\}_{k_{e_u}}$ unto ECU e.

In Fig. 3, e_9 sends requesting message to join. And it is assumed that the approval for e_9 the secure group. The server creates new group key $k_{e_{1-9}}$ and

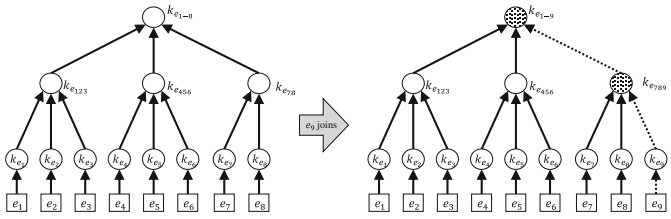

Fig. 3. ECU e_9 requests to join in a tree key graphs

new sub-group key k_{e789} which call joining point node's key. Among the existing ECU $e_1,\ e_2, \ldots,\ e_6$ need to new group key (i.e., $k_{e_{1-9}}$) and $e_7,\ e_8$ need to a new group key, a new sub-group key (i.e., $k_{e789}, k_{e_{1-9}}$). The server sends securely rekey message to distribute the key. Rekey message is as follows.

$$s \rightarrow \{e_1, e_2, \ldots, e_8\} : E_{k_{e_{1-8}}}[k_{e_{1-9}}],\ E_{k_{e78}}[k_{e789}] \cdots \quad (1)$$

$$s \rightarrow\ e_9 \qquad\qquad : E_{k_{e_9}}[k_{e_{1-9}}, k_{e789}] \cdots\cdots\cdots\cdots \quad (2)$$

The server sends (1) message for $e_1,\ e_2, \ldots,\ e_8$ to multicast and sends (2) message for e_9 to unicast. If each ECU received messages, they would have only necessary information and discard unnecessary information. In this consist of rekey message, since a number of rekey message minimizes, overhead of the server reduced.

Leaving. An ECU e which wants to leave the secure group sends requesting message to leave to the key distribution server. If join request is approved, the server deletes *e-node* of ECU e and *k-node* of e's individual key. This *k-node*'s parent called leaving point. To prevent the access of the leaving member, keys must be changed from leaving point to root node. The server creates new group key and distributes securely to the remaining members.

If the server responds message to leave, that thing as follows. The server searches for a parent node of leaving e's individual key k_{e_u} which is called leaving point. And remove k_{e_u} from the tree. x_{j+1} denote the deleted *k-node* for k_{e_u}, x_j the leaving point, x_0 the root, and when $i = 1, \ldots, j,\ x_{i-1}$ the parent of K'_0. The server generates randomly keys $K'_0, \ldots, K'_j$ as the new keys of $x_0, \ldots, x_j$. And when $i = 0, \ldots, j,\ J_1, \ldots, J_r$ denote key at the children of x_i in the new tree key graphs. The server encrypts K'_i to each children key that called L_i. This denotes $\{K'_i\}_{J_1}, \ldots, \{K'_i\}_{J_r}$. When expressed as a function, it is $ECUset(K) : L_0, \ldots, L_j$.

In Fig. 4, e_9 sends requesting message to leave. And it is assumed that the approval for e_9 the secure group. The server creates new group key $k_{e_{1-8}}$ and new sub-group key $k_{e_{78}}$ which called leaving point node's key. Among the existing ECU $e_1,\ e_2, \ldots,\ e_6$ need to a new group key (i.e., $k_{e_{1-8}}$) and $e_7,\ e_8$ need to a new group key, a new sub-group key (i.e., $k_{e_{78}}, k_{e_{1-8}}$). The server sends securely rekey message to distribute the new key. Rekey message is as follows.

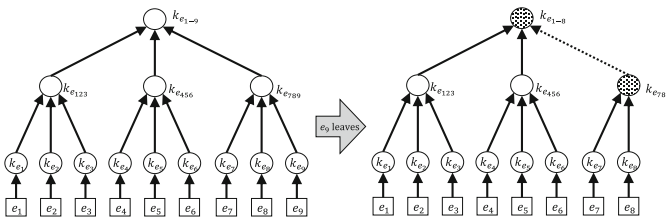

Fig. 4. ECU e_9 requests to leave in a tree key graphs

Let L_0 denote $E_{k_{e_{123}}}[k_{e_{1-8}}],\ E_{k_{e_{456}}}[k_{e_{1-8}}],\ E_{k_{e_{78}}}[k_{e_{1-8}}]$
Let L_1 denote $E_{k_{e_7}}[k_{e_{78}}],\ E_{k_{e_8}}[k_{e_{78}}]$

$$s \rightarrow \{e_1, e_2, \ldots, e_8\} : L_0, L_1 \cdots\cdots\cdots\cdots\cdots \quad (3)$$

This approach uses only one rekey message. The server sends (3) message for $e_1,\ e_2, \ldots,\ e_8$ to multicast. A rekey message is configured to include all keys.

4 Conclusion

In this paper, we have presented a group key based session key establishment protocol for a remote vehicle diagnostic. The proposed scheme aims at providing secure authentication and session key establishment between a vehicle manufacturer server and a group of in-vehicle electronic control unit based on a key graph. The proposed scheme has better key management efficiency than symmetric key systems, while providing lower computation cost than public key systems.

Acknowledgment. This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (NRF-2014R1A1A1006770).

References

1. Sangguk, K.: Connected car drastic market expectations in Iot/M2M technological environment. In: KISTI (Korea Institute of Science and Technology Information), February 2014
2. Greenough, J.: The connected car report: forecasts, competing technologies, and leading manufacturers. BI Intell. (2016)
3. Miller, C., Valasek, C.: Remote exploitation of an unaltered passenger vehicle. In: DEFCON, August 2015
4. Wi, S., Moon, K., Lee, J.-H.: Symmetric key based session key establishment protocol for remote ECU management. In: KIISE (Korean Institute of Information Scientists and Engineers), December 2015
5. Gang, S.: Connected car development trends and changes of the future. KB Finance Support Management Laboratory, February 2016
6. Vehicle ECU Analyzing & Market report 2012, Fuji Chimera Research Institute, February 2012
7. 2013 ETRI technology preview, ETRI (Electronics and Telecommunications Research Institute) (2013)
8. CAN - CiA. http://www.can-cia.org/can-knowledge/can/high-speed-transmission/
9. Flexray - Embedded System Korea. http://www.eskorea.net/html/data/technique/ixFlexray.pdf
10. Tuohy, S., Glavin, M., Hughes, C., Jones, E., Trivedi, M.M, Kilmartin, L.: Intra-vehicle network: a review. IEEE Trans. Intell. Transp. Syst. (2015)
11. Secure requirements and framework for multicast communication. TTA (Telecommunications Techology Association) (2010)

12. Wong, C.K., Gouda,M., Lam, S.S.: Secure group communications using key graphs. IEEE Commun. Soc. (2000)
13. Kumar, P., Lee, S.-G., Lee, H.-J.: E-SAP: Efficient-Strong Authentication Protocol for healthcare applications using wireless medical sensor networks. MDPI, December 2012