Symmetric-key Inter-domain end-to-end Authentication Protocols for Mobile System

Rajarshi Roy Chowdhury[1], Subramanian Azhaguvel[2]
[1]Final Year Computer Science, INTI College ,Kuching (University of Wollongong), Sarawak, Malaysia
[2]Lecturer, INTI College , Kuching (University of Wollongong), Sarawak, Malaysia

Abstract

Our goal is to propose and design a symmetric-key inter-domain end-to-end authentication protocols for a mobile system. We will be considering three mobile service domains; each has an authentication server. We denote by AS1,AS2 and AS3 corresponding authentication servers. For simplicity, let AS1, AS2 ,AS3 represent those three domains. This mobile system can provide mobile communication services to a large number of users. For simplicity, we assume three mobile users (A, B and C) in the system only, where A has registered with AS1 and B has registered with AS2 and C has registered with AS3.

Keywords : Symmetric-key, inter-domain, authentication protocols, authentication server, mobile communication.

I. INTRODUCTION

A) Symmetric-key

An encryption system in which the sender and receiver of a message share a single, common key that is used to encrypt and decrypt the message. Contrast this with public-key cryptology, which utilizes two keys - a public key to encrypt messages and a private key to decrypt them. Symmetric-key systems are simpler and faster, but their main drawback is that the two parties must somehow exchange the key in a secure way. Public-key encryption avoids this problem because the public key can be distributed in a non-secure way, and the private key is never transmitted. Symmetric-key cryptography is sometimes called *SECRET-KEY CRYPTOGRAPHY*. The most popular symmetric-key system is the *DATA ENCRYPTION STANDARD (DES*). [1]

B) Types

Symmetric-key algorithms can be divided into stream ciphers and block ciphers. Stream ciphers encrypt the bits of the message one at a time, and block ciphers take a number of bits and encrypt them as a single unit. Blocks of 64 bits have been commonly used. The Advanced Encryption Standard (AES) algorithm is approved by NIST where uses 128-bit blocks are. Examples of some symmetric algorithms include Twofish, Serpent, AES (Rijndael), Blowfish, CAST5, RC4, TDES, and IDEA.

C) Security issue

Symmetric ciphers have historically been susceptible to known-plaintext attacks, chosen plaintext attacks, differential cryptanalysis and linear cryptanalysis. Careful construction of the functions for each round can greatly reduce the chances of a successful attack

D) Inter Domain

In computing, **inter-domain** is a term used to describe interaction between domains. It is most commonly used in the fields of multicasting and routing between internets, or as a substitute for the term inter-server. Internet protocols that are focused on inter-domain functions include: Border Gateway Multicast Protocol, Classless Inter-Domain Routing, Multicast Source Discovery Protocol, and Protocol Independent Multicast. The opposite of inter-domain routing is intra-domain routing (routing within a domain or an autonomous system).

E) End-To-End Authentication

An authentication protocol is a type of cryptographic protocol with the purpose of authenticating entities wishing to communicate securely. There are many different authentication protocols such as: Kerberos, RADIUS (Remote Authentication Dial In User Service) and so on. End to end authentication protocol is a computer network authentication protocol, which allows nodes communicating over a non-secure network to prove their identity to one another in a secure manner. It provides mutual authentication — both the user and the server verify each other's identity.[2]

II. PROPOSED PROTOCOL DESIGN

A) Notations

A, B, C: End-users (where A, B and C are mobile nodes).
$\mathbf{A_S}$: Subliminal identity of A.

$\mathbf{B_S}$: Subliminal identity of B.
$\mathbf{C_S}$: Subliminal identity of C.
$\mathbf{A_S'}$: New subliminal identity of A.
$\mathbf{B_S'}$: New subliminal identity of B.
$\mathbf{C_S'}$: New subliminal identity of B.
AS1 : Home domain server of user A, and also the foreign domain server of user B and C.
AS2 : Home domain server of user B, and also the foreign domain server of user A and C.
AS3 : Home domain server of user C, and also the foreign domain server of user A and B.
$\mathbf{K_S}$: Secret session key used by A, B and C to communicate securely.
$\mathbf{K_{A, AS1}}$: Shared secret key between A and AS1.

$\mathbf{K_{B, AS1}}$: Shared secret key between B and AS1.
$\mathbf{K_{AS1, AS2}}$: Shared secret key between AS1 and AS2.
$\mathbf{K_{AS1, AS3}}$: Shared secret key between AS1 and AS3.
$\mathbf{K_{B, AS2}}$: Shared secret key between B and AS2.
$\mathbf{K_{C, AS3}}$: Shared secret key between C and AS3.
$\mathbf{[data]_{key}}$: data encrypted with the symmetric key.
h (…) : A strong one-way hash function.
$\mathbf{n_A}$: Nonce generated by A.
$\mathbf{n_B}$: Nonce generated by B.
$\mathbf{n_C}$: Nonce generated by C.
$\mathbf{nA'}$: New nonce generated by A.
$\mathbf{n_B'}$: New nonce generated by B.
$\mathbf{n_B''}$: New nonce generated by B.
$\mathbf{n_C'}$: New nonce generated by C.
$\mathbf{n_{AS1}'}$: New nonce generated by AS1.
$\mathbf{n_{AS1}''}$: New nonce generated by AS1.
$\mathbf{n_{AS2}'}$: New nonce generated by AS2.
$\mathbf{n_{AS3}'}$: New nonce generated by AS3.
A −> **B**: *message:* This means A sends message to B.
A −> **C**: *message:* This means A sends message to C.
B −> **A**: *message:* This means B sends message to A.
B −> **C**: *message:* This means B sends message to C.
C −> **B**: *message:* This means C sends message to B.
C −> **A**: *message:* This means C sends message to A.

B) Protocol Illustration

(With Authenticity, Confidentiality, Anonymity, and Freshness)

STEP 1: B −> AS1 *: B_S, AS2, n_B, $Token_{B,AS1,AS2}$, [B, B_S, A, C] $K_{B,AS1}$, [h(B_S, AS2, n_B)]$K_{B,AS1}$*

Where, $K_{A,AS2} = f(K_{B,AS2}, B_S, AS1)$,
$Token_{B,AS1,AS2} = [B, AS1, AS2, n_B]K_{A,AS1}$

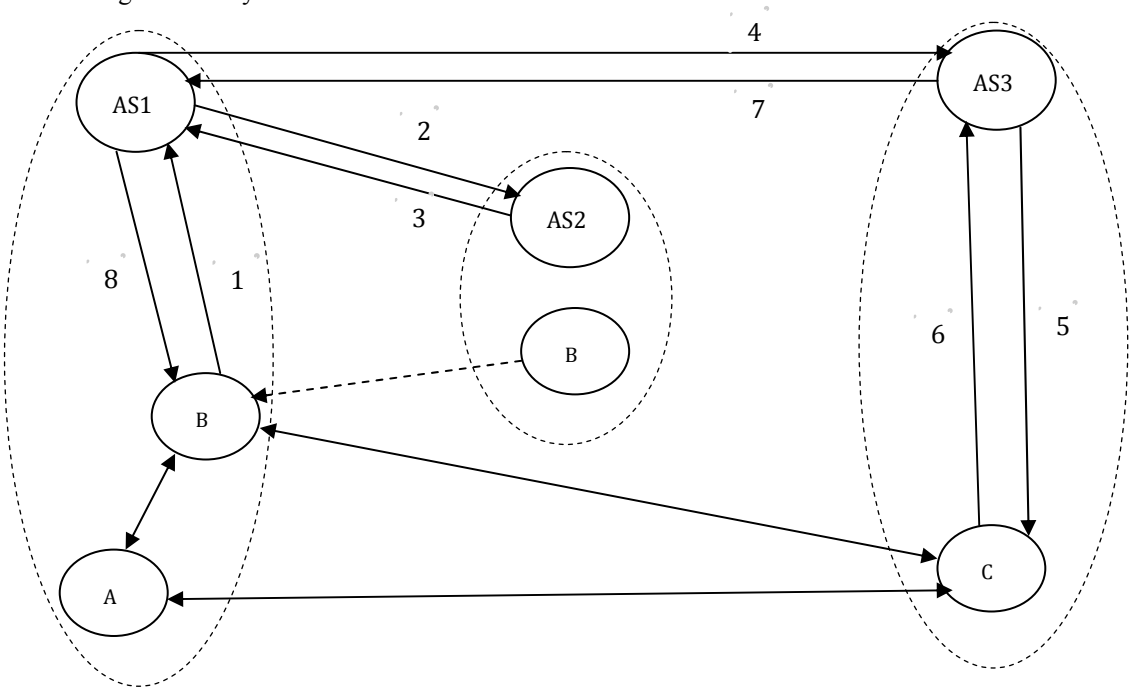

Figure 1.4: A and C their home Domain and B move to A's home domain

STEP 2: AS1 *−> **AS2**: AS1, AS2, n_{AS1}, B_S, $Token_{B, AS1, AS2}$, [h (AS1, AS2, n_{AS1}, B_S, $Token_{B,AS1,AS2}$)]* $K_{AS1,AS2}$

STEP3: AS2 *−> **AS1**: AS2, AS1, n_{AS1}, [$K_{B,AS1}$, B_S]* $K_{AS1,AS2}$, *[h (AS2, AS1, n_{AS1}, $K_{B,AS1}$, B_S)]* $K_{AS1,AS2}$,*[B_S']* $K_{B,AS2}$, *[h(B_S', AS2, n_B)]*$K_{B,AS2}$

STEP 4: AS1 *−> **AS3**: AS1, AS3, n_{AS1}', [B, B_S, A, A_S, n_B, C, K_S]* $K_{AS1,AS3}$, *[h (AS1, AS3, n_{AS1}', B, B_S, A, A_S, n_B, C, K_S)]* $K_{AS1,AS3}$

STEP 5: AS3 *−> **C**: AS3, C_S, $n_{AS\ 3}$, [B, B_S, A, A_S, n_B, C, C_S' K_S]* $K_{AS3,B}$, *[h (AS3, C_S, $n_{AS\ 3}$, B, B_S, A, A_S, n_B, C, C_S' K_S)]* $K_{AS3,B}$

STEP 6: C *−> **AS3**: C_S, AS3, $n_{AS\ 3}$, [h (C_S, AS3, $n_{AS\ 3}$, B, A,n_B, C,K_S)]* $K_{AS3,B}$

STEP 7: AS3 *−> **AS1**: AS3, AS1, n_{AS1}', [C_S]* $K_{AS3,AS1}$, *[h (AS3, AS1, n_{AS1}', C_S, B,A, n_B, C, K_S)]* $K_{AS3,AS1}$

STEP 8: AS1 *−> **B** : AS1, B, n_B, n_{AS1}'', [K_S, B, B_S, C, C_S]* $K_{B,AS1}$,*[h (AS1, B, n_B, n_{AS1}'',K_S, B, B_S, C, C_S)]* $K_{B,AS1}$, *[C, C_S, B, B_S, K_S]* $K_{A,AS1}$, *[h(C, C_S, B,B_S, K_S, AS1)]*$K_{A,AS1}$,*[A_S']* $K_{A,AS1}$, *[h(A_S', AS1, n_A)]*$K_{A,AS1}$,*[B_S']* $K_{B,AS2}$, *[h(B_S', AS2, n_B)]*$K_{B,AS2}$

STEP 9: B *−> **A** : B_S, A_S, AS1, n_B', [B_S, A_S, AS1, n_B']* K_S,*[C, C_S, B, B_S, K_S]* $K_{A,AS1}$, *[h(C, C_S, B, B_S, K_S, AS1)]*$K_{A,AS1}$,*[A_S']* $K_{A,AS1}$, *[h(A_S', AS1, n_A)]*$K_{A,AS1}$,

STEP 10: A *−> **B** : A_S, B_S, [message, n_B']* K_S

STEP 11: B *−> **C** : B_S, C_S, [message, n_B'']* K_S

STEP 12: C *−> **B** : C_S, B_S, [message, n_B'']* K_S

STEP 13: A *−> **C** : A_S, C_S, [message, n_A]* K_S

STEP 14: C *−> **A** : C_S, A_S, [message, n_A]* K_S

III. PROTOCOL DESCRIPTION

STEP 1: B *−> **AS1**:B_S, AS2, n_B, $Token_{B,AS1,AS2}$, [B, B_S, A, C]* $K_{B,AS1}$, *[h(B_S, AS2, n_B)]*$K_{B,AS1}$

Where, $K_{A,AS2} = f(K_{B,AS2}, B_S, AS1)$,
$Token_{B,AS1,AS2} = [B, AS1, AS2, n_B]K_{A,AS1}$

a) **B** sends **AS1** a request, including a Token, nonce, "identity package" and hash value. **B**'s subliminal identity for **anonymity**.

b) The Token is encrypted so as to be passed on to **AS2** without **AS1** being able to read it.

c) The content of the Token will allow **AS2** to authenticate **B**.

d) **AS1** cannot verify the hash value or decrypt the request, since it doesn't have $K_{B,AS1}$, which is generated using a strong one-way hash function **f**.

e) Only **B** and **AS2** can calculate $K_{B,AS1}$.

STEP 2: AS1 *−> **AS2**: AS1, AS2, n_{AS1}, B_S, $Token_{B,AS1,AS2}$,[h (AS1, AS2, n_{AS1}, B_S, $Token_{B,AS1,AS2}$)]* $K_{AS1,AS2}$

a) After receiving the Token, the Home server **AS2** is able to **authenticate** B.

STEP 3: AS2 *−> **AS1**: AS2, AS1, n_{AS1}, [$K_{B,AS1}$, B_S]* $K_{AS1,AS2}$, *[h (AS2, AS1, n_{AS1}, $K_{B,AS1}$, B_S)]* $K_{AS1,AS2}$, *[B_S']* $K_{B,AS2}$, *[h(B_S', AS2, n_B)]*$K_{B,AS2}$

a) **AS2** sends a new subliminal identity.

b) The identity and second hash value will be passed to **B**.

c) **AS2** gives the key $K_{B,AS1}$ and **B**'s subliminal identity.

d) **AS1** can use this to verify the hash value received from B_S in the first step.

e) **AS1** now knows who B_S wants to talk to.

STEP 4: AS1 *−> **AS3**: AS1, AS3, n_{AS1}', [B, B_S, A, A_S, n_B, C, K_S]* $K_{AS1,AS3}$, *[h (AS1, AS3, n_{AS1}', B, B_S, A, A_S, n_B, C, K_S)]* $K_{AS1,AS3}$

a) Upon verification of the request, **AS1** generates a secret session key K_S, which is for distribution to **A** and **C**.

b) **AS1** passes the subliminal identity to **AS2**, otherwise **C** at far end won't be satisfied that **A** and **B** are trying to communicate with them later.

STEP 5: AS3 $-> C$: $AS3, C_S, n_{AS3}, [B, B_S, A, A_S, n_B, C, C_S' K_S] K_{AS3,B}, [h (AS3, C_S, n_{AS3}, B, B_S, A, A_S, n_B, C, C_S' K_S)] K_{AS3,B}$

a) **AS3** passes the secret session key $\mathbf{K_S}$, along with **A**'s, **B**'s and **C**'s identities and the nonce $\mathbf{n_B}$, along to **C** encrypted under $\mathbf{K_{AS3,B}}$.

b) Authentication of **B** to **C** is complete.

c) **AS3** also send an updated subliminal identity (for **C**) at this stage.

STEP 6: C $-> AS3$: $C_S, AS3, n_{AS3}, [h (C_S, AS3, n_{AS3}, B, A, n_B, C, K_S)] K_{AS3,B}$

a) Start authentication of C to B.
C sends **AS3** the hash value containing the secret session key $\mathbf{K_S}$, nonce(s) and **A**'s, **B**'s and **C**'s identities, all encrypted under $\mathbf{K_{AS3,B}}$.

STEP 7: AS3 $-> AS1$: $AS3, AS1, n_{AS1}', [C_S] K_{AS3,AS1}, [h (AS3, AS1, n_{AS1}', C_S, B, A, n_B, C, K_S)] K_{AS3,AS1}$

a) Upon verification of the hash value, **AS1** is aware of whether or not **C** has received the correct session key, and whether the information is fresh.

b) The subliminal identity of **C** is being passed back for **B** and **A** to use.

STEP 8: AS1 $-> B$: $AS1, B, n_B, n_{AS1}'', [K_S, B, B_S, C, C_S] K_{B,AS1}, [h (AS1, B, n_B, n_{AS1}'', K_S, B, B_S, C, C_S)] K_{B,AS1}, [C, C_S, B, B_S, K_S] K_{A,AS1}, [h(C, C_S, B, B_S, K_S, AS1)] K_{A,AS1}, [A_S'] K_{A,AS1}, [h(A_S', AS1, n_A)] K_{A,AS1}, [B_S'] K_{B,AS2}, [h(B_S', AS2, n_B)] K_{B,AS2}$

a) It would appear that **AS1** sends everything it can find.

b) It distributes the session key $\mathbf{K_S}$, encrypted under $\mathbf{K_{B,AS1}}$ for **B** and under $\mathbf{K_{A,AS1}}$ for **A**.

c) **B** receives its new subliminal identity for use in the future communication.

d) **AS1** also send **A**'s new subliminal identity.

STEP 9: B $-> A$: $B_S, A_S, AS1, n_B', [B_S, A_S, AS1, n_B'] K_S, [C, C_S, B, B_S, K_S] K_{A,AS1}, [h(C, C_S, B, B_S, K_S, AS1)] K_{A,AS1}, [A_S'] K_{A,AS1}, [h(A_S', AS1, n_A)] K_{A,AS1}$,

a) **B** distributes the session key $\mathbf{K_S}$, encrypted under $\mathbf{K_{A,AS1}}$ for **A**.

b) **A** receives its new subliminal identity for use in the future communication.

STEP 10: A $-> B$: $A_S, B_S, [message, n_B'] K_S$

a) **A** sends message to **B** with nonce and under encrypted with session key $\mathbf{K_S}$.

STEP 11: B $-> C$: $B_S, C_S, [message, n_B''] K_S$

a) **B** sends message to **C** with nonce and under encrypted with session key $\mathbf{K_S}$.

STEP 12: C $-> B$: $C_S, B_S, [message, n_B''] K_S$

a) C sends message to B with nonce and under encrypted with session key K_S.

STEP 13: A $-> C$: $A_S, C_S, [message, n_A] K_S$

a) A sends message to C with nonce and under encrypted with session key K_S.

STEP 14: C $-> A$: $C_S, A_S, [message, n_A] K_S$

a) **C** sends message to **A** with nonce and under encrypted with session key $\mathbf{K_S}$.

IV. ADVANTAGES OF PROTOCOL

There are some key advantages using propose Inter Domain protocol. Advantages are listed below:

1. Using subliminal user identity (called anonymity).
2. Nonce – using for anti reply attack.
3. Provides data integrity and confidentiality.
4. Users are authenticated by their home domain.
5. Establishment session key for particular session.
6. Updated subliminal identity after each session by the home server.

V. EXPERIMENTAL ANALYSIS

A) Case 1

There are three mobile service domains; each has an authentication server. The authentication servers are denoted by AS1, AS2 and AS3. Assume there are three mobile users A, B and C accordingly registered
with AS1, AS2 and AS3. If they want to communicate securely with each other then they have to follow the steps below (conference call).

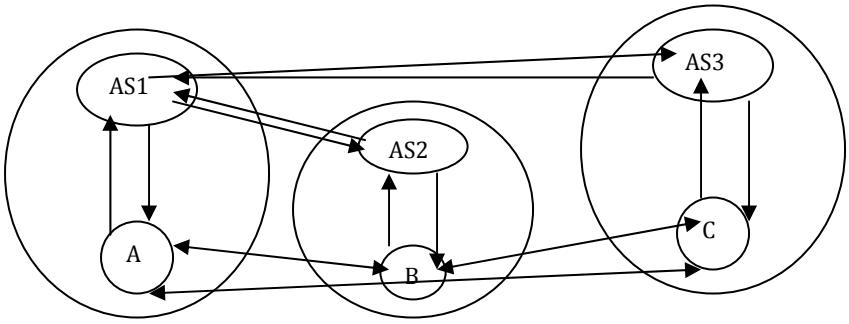

Figure 1.4: A, B, and C in their home Domain

Step 1: A makes a request to AS1, he wants to talk to B and C.

Step 2: First AS1 is looking for B and C location, or which authentication server they are belongs to. Then AS1 send a request to B's home server with A's identity, nonce and secret session key.

Step 3: AS2 now talk to B that A wants to talk to you and send A's subliminal identity, nonce, and secret key to B. Send B's new identity by the AS2.

Step4: B confirms to AS2 he is ready to communicate with A, and send his identity along with session key encrypted with $K_{B,AS2}$.

Step5: Then AS2 send B's subliminal identity to the AS1, and complete B's authenticity.

Step6: AS1 now send a request to AS3, along with A's and B's identity, nonce and session key for secure communication.

Step 7: AS3 now talk to C that A and B wants to talk to you and send A's and B's subliminal identity, nonce, and secret key to C. AS3 also send C's new identity.

Step 8: C confirms to AS3 he is ready to communicate with A and B, and send his identity along with session key encrypted with $K_{C,AS3}$.

Step 9: Then AS3 send C's subliminal identity to the AS1, and complete C's authenticity.

Step 10: AS1 now sends all he has to A.

Step 11: Now A, B and C can communicate securely.

B) Case 2

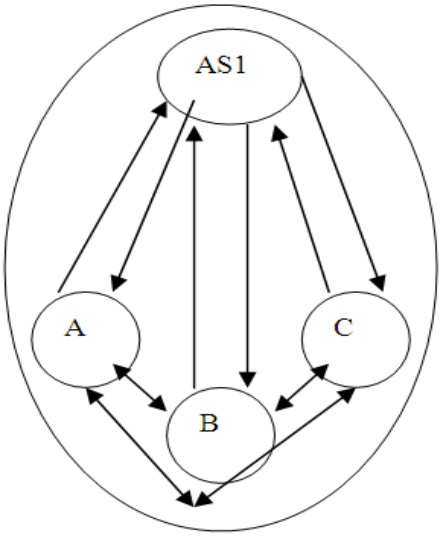

Figure 1.4: A, B, and C in the same Domain

There is only one mobile service domain and has an authentication server. The authentication server is denoted by AS1. Assume there are three mobile users A, B and C, each one is registered with AS1. If A, B and C wants to communicate securely with each other then they have to follow the steps below (conference call).

Step 1: A send a request to his home (AS1) server that A wants to communicate with B and C.

Step 2: After verifying request AS1 is looking for B and C's home domain and determine that B and C are in the same home domain. Now AS1 generate a session key K_{ABC} and send to B along with B's new identity for future communication.

Step 3: Now B confirms to AS1 that B is ready to communicate with A and C.

Step 4: AS1 talks to C that A and B want to talk to you; then AS1 sends session key K_{ABC} along with new identity of C.

Step 5: C confirms to AS1 that C is ready to communicate with A and B, and send identity along with response.

Step 6: AS1 send session key with identity of both B and C to A and A's new identity.

Step 7: A, B and C can communicate securely.

VI CONCLUSION

We are all aware of the growth in routing complexity, and the rapid increase in allocation of network numbers. So, we need some setup rules for secure communication between end-to-end machines. In Inter-Domain Routing Protocol (IDRP) provides secure routing for OSI defined network environments, which is similar to BGP in the TCP/IP network. The Border Gateway Protocol (BGP) provides a standard mechanism for inter-domain routing among heterogeneous domains, called autonomous systems (AS), where each domain has the administrative control over its intra-domain routing protocol and inter-domain routing policy, which is not known to the other domains.

ACKNOWLEDGMENTS

We wish to thank the editor and the anonymous reviewers for their constructive comments and detailed suggestions to improve the paper's presentation. We thank the faculty of INTI College Kuching, Sarawak for helpful discussions on the materials presented in this paper.

REFERENCES

[1] http://www.webopedia.com/TERM/S/symmetric_key_cryptography.html
[2] http://en.wikipedia.org/wiki/Inter-domain
[3]http://www.freepatentsonline.com/7240366.html
[4]https://wiki.internet2.edu/confluence/download/attachments/19074/IDC-Messaging-draft.pdf
[5] http://en.wikipedia.org/wiki/Anonymity
[6]http://en.wikipedia.org/wiki/Cryptographic_nonce
[7]http://www.networkdictionary.com/protocols/idrp.php
[8]http://en.wikipedia.org/wiki/Session_key
[9]http://en.wikipedia.org/wiki/Universal_one-way_hash_function
[10]http://www.ist-intermon.org/download/IM-WP3-FOKUS-CINI-Interdomain_Issues-draft.pdf

BIOGRAPHY

RAJARSHI ROY CHOWDHURY Final Year Degree Student currently doing his Bachelor of Computer Science in Digital Systems Security at INTI College , Kuching (Offshore Campus of University of Wollongong, Australia), Sarawak, Malaysia. My area of interest is Mobile Computing and Computer Networks.

Subramanian Azhaguvel received the Bachelor of Computer Science Engineering degree from Madurai Kamaraj University, TN, India and the Master of Computer Science Engineering degree from Anna University, Chennai, TN, India. He is currently doing his part time Research at the Jawaharlal Nehru Technological University, Hyderabad, Andhra Pradesh and presently working as a Lecturer in INTI College, Kuching, Sarawak, Malaysia. His research interests include Network Security, Computer Networks, Intelligent systems, Data Mining, Bioinformatics, and Computational Web-intelligence.

Mr. Subramanian Azhaguvel received the Young Scientist Award in 2007. He is a member in Computer Society of India.