SSL-MAP: A More Secure Gossamer-based Mutual Authentication Protocol for Passive RFID Tags

Rama N.
Department of Computer Science
Presidency College
Chennai, India
n_ramabalu@yahoo.com

Suganya R.
Department of Computer Science, Meenakshi College for Women, Chennai, India & Research Scholar, Mother Teresa Women's University, Kodaikanal, India
suganyaramadoss@gmail.com

***Abstract*— RFID systems that employ passive RFID tags, are run using lightweight protocols. The Gossamer protocol is a case in point. However, it is found that the Gossamer protocol uses rather simple operations, in order to ensure that the protocol is lightweight. This raises security concerns.**

A protocol based on the Sign/Logarithm number system to make it power-efficient, and the efficient use of one-dimensional convolution to secure the transmission of the tag's ID value, is proposed. Thus, a protocol that is power-efficient and more secure than the Gossamer protocol is proposed. Further, this protocol may be used even in the basic passive tag, which has minimal processing capability and no power source of its own.

Keywords- RFID, Gossamer, Sign/Logarithm number system, mutual authentication protocol, Convolution

I. INTRODUCTION

Radio Frequency IDentification (RFID) systems are being increasingly adopted for use in domains as diverse as supply chain management and high security zones of the military.

An RFID system has RFID tags and readers. The communication channel between the tag and the reader is insecure. Hence security has to be guaranteed by the protocol used for the message-exchange between the tag and the reader. As the RFID tags are a resource-constrained environment, traditional cryptographic algorithms that can provide high levels of security, have been found unsuitable for implementation in RFID tags until now.

The Gossamer protocol is a mutual authentication protocol that belongs to the Ultra-lightweight mutual authentication protocols (UMAP) family [10]. These protocols [6, 7, 8, 4] were inspired by the scheme called minimalist cryptography proposed by Ari Juels [1]. A new family of twin protocols based on the Gossamer protocol, has also been proposed by the authors for the tags to mutually authenticate each other [12].

An enhanced, secure mutual authentication protocol, SSL-MAP (Secure Sign/Logarithm-based Mutual Authentication Protocol) is now proposed, which provides significant improvements over the Gossamer protocol in terms of performance and security.

II. RFID TAGS

A. Passive RFID Tags

Passive RFID tags have a low-power integrated circuit (IC) and an attached antenna. The IC has an onboard memory which stores data, and uses the antenna to transmit signals to communicate with the RFID reader. The structure of the antenna differs, depending on whether the tag is a high-frequency (HF) tag or an ultra high-frequency (UHF) one [2].

Passive RFID tags do not have a power source of their own. They are powered only by the energy transferred from the reader through an energy-coupling mechanism, and are active only when they are within the field of the reader. The signal from the reader also energizes the tag, apart from communicating with it. Passive RFID tags typically have a communication range of only about 3 meters. [3, 15]

B. EPC Global Classification of RFID tags

The four classes of RFID tags identified by EPC Global [16] are:

- Class 1 – Identity tags
  Passive-backscatter tags with an EPC identifier, a tag identifier (TID), a kill function that permanently disables the tag, optional password-protected access control, and optional user memory.
- Class 2 – Higher functionality tags
  Passive tags with all the aforementioned features with an extended TID, extended memory and authenticated access control.
- Class 3 - Semi-passive tags
  Besides the above features these have an integral power source and integrated sensing circuitry.
- Class 4 - Active tags
  In addition to the above, these are equipped with tag-to-tag communication, active communication and ad-hoc and networking capabilities.

III. THE GOSSAMER PROTOCOL – AN OVERVIEW

Gossamer protocol is a mutual authentication protocol designed for EPC C1G2 tags [10]. This is a protocol in the family of ultra-lightweight mutual authentication protocols. The phases involved in this protocol are (i) Tag identification (ii) Mutual authentication and (iii) Index pseudonym (IDS) and key updating. The last phase involves updates to the IDS and the pair of keys unique to each tag. This update is done for every run of the protocol, independently by the reader and the tag, using the same set of equations.

The operations that are employed in this protocol are

- Concatenation represented by ||
- XOR represented by ⊕
- Addition modulo 2^m represented by +
- ROT(x, y) defined as circular left shift of x by (y mod 96)
- Bitwise right shift represented by >>

The Gossamer protocol uses a specially designed lightweight function named *MixBits*, having its base in genetic programming, to generate more random numbers from the two random numbers generated by the reader in each run [9]. The value of m involved in addition modulo 2^m can be 4, 8 or 16.

IV. The Need for an Enhanced Protocol

The Gossamer protocol extensively employs the shift and modulo operations while generating messages. Though these are by far simple operations, they consume considerable amounts of power. A power analysis of the Gossamer protocol with special reference to the shift operations was undertaken by the authors and the results presented in earlier work [13], detailing the impact of the shift operations on the power consumed by the protocol.

The modulo operator too is extensively used in the Gossamer protocol. The number of subtractions involved in calculating the remainder plays a major role in the performance and power consumption of the protocol. In an attempt to reduce the computations involved in this operation, the Sign/Logarithm number system was employed by the authors to arrive at an enhanced power-efficient Gossamer-based protocol [14].

Now the reason for only the simple shift and modulo operations being used in the Gossamer protocol, is that the protocol necessarily has to be lightweight because it is applied in a hugely resource-constrained environment, viz. the passive tag. Though the Gossamer protocol is strong enough to have not been broken till now in the resource-constrained environment in which it is applied, a cryptanalysis carried out with larger computing power would certainly show that an attack is possible. This is obvious from the simplicity of the operations involved in the protocol.

As already shown [13, 14], even the shift and modulo operations employed consume huge amounts of power. Drastically reducing the power consumption of these operations through the use of the Sign/Logarithm number system, was also shown to be a feasible alternative [14]. Hence, it is now proposed to secure the protocol further by using a non-traditional cryptographic primitive within the framework of the Sign/Logarithm number system. It is seen that the power saved through the use of the Sign/Logarithm number system is sufficient to accommodate the operation of one-dimensional convolution to reap significantly larger security benefits.

A. An overview of the Sign/Logarithm Number System

In this number system, a number is represented by a sign bit and the logarithm of the absolute value of the number (scaled to avoid negative logarithms) [5].

A number A is represented by its sign S_A and the binary logarithm L_A of its value, where

$$S_A = 1 \qquad if\ A \leq 0 \tag{1}$$

$$S_A = 0 \qquad if\ A \geq 0 \tag{2}$$

$$L_A = \log(|\tau A|) \qquad if\ |A| > 1/\tau \tag{3}$$

$$L_A = 0 \qquad if\ |A| \leq 1/\tau \tag{4}$$

The following equations are used to convert a number A to its corresponding Sign/Logarithm form:

$$K_A = 2^{1-\eta}\,[\tfrac{1}{2} + 2^{\eta-1}\log_2|\tau A|] \qquad if\ |A| > 1/\tau \tag{5}$$

$$K_A = 0 \qquad if\ |A| \leq 1/\tau \tag{6}$$

where K_A is the finite precision form of L_A, η-1 is the number of decimal places to be used in the system being designed, τ is the scaling factor chosen in order to avoid negative logarithms and [X] denotes the largest integer that is not larger than X. A log lookup table is created using the equations (5) and (6), which plays a major role in reducing the complexity of multiplication and division operations. The steps needed to compute the result are fixed, irrespective of the numbers involved in the operation [5].

B. Application of the Sign/Logarithm number system to the Gossamer Protocol

This section explains with examples the calculation of product and modulo using the Sign/Logarithm number system and thus establishes how the number of steps remains constant in these operations, irrespective of the operands. The examples given below are with log10, although log2 is preferred in practice because it conduces to a reduced table size.

Calculating the product of two numbers:

Let N = 12345 * 254 = 12.345 * 25.4 * 10^4

log N = log 12.345 + log 25.4 + 4

N = antilog (log 12.345 + log 25.4 + 4) (7)

Generalizing, equation (7) for P = N *M, where the i-digit number N is represented by q1q2q3...qi, 0 <= qk <= 9, 1 <= k <= i and M is represented by r1r2r3...rj, 0 <= rk <= 9, 1 <= k<= j, equation (8) is obtained.

$$P = \text{antilog}\,(\log\,(q_1q_2.q_3...q_i) + \log\,(r_1r_2.r_3...r_i) + i + j - 4) \tag{8}$$

Thus to compute product of any two numbers the operations involved are two additions and three memory accesses – two for the log value and the other for the antilog. (Addition and subtraction operations denote binary addition and binary subtraction respectively.)

Calculating Modulo:

Let Q = 8674532 / 255.

log (Q) = log (8674532/255)

= log (8674532) – log (255)

= log (86.74532 * 10^5) – log (25.5*10)

= 4 + log (86.74532) – log (25.5) (9)

Let R = 8674532 mod 255

= 8674532 – (255 * Q)

= 8674532 – antilog (log (255) + log(Q)) (10)

Substituting from equation (9) in equation (10),

R = 8674532 – antilog (log (255) + 4 + log (86.74532) – log (25.5))

R = 8674532 – antilog (5 + log (86.74532)) (11)

Generalized equation for $R = N$ mod M, where the i-digit number N is represented by $q_1q_2q_3...q_i$, 0 <= q_j <= 9, 1 <= j <= i, is given by

$R = N$ – antilog (i - 2 + log ($q_1q_2.q_3...q_i$)) (12)

To compute equation (12) the operations involved are one addition, one subtraction and two memory accesses – one for the log value and the other for the antilog.

Table 1 summarizes the number of operations involved in the computation of product and remainder using Sign/Log number system.

TABLE I. SUMMARY OF THE OPERATIONS INVOLVED

Operation	Number of operations involved
Multiplication	• 3 memory lookups • 2 additions
Modulo	• 2 memory lookups • 1 addition • 1 subtraction

Modulo operations are extensively used in the Gossamer protocol. Hence employing the Sign/Logarithm based number system will improve the performance of the protocol and reduce the power consumed.

V. SSL-GMAP: A SECURE, POWER-EFFICIENT GOSSAMER-BASED LIGHTWEIGHT MUTUAL AUTHENTICATION PROTOCOL

The proposed protocol has the following three steps (i) Tag identification (ii) Mutual Authentication (iii) IDS and key updating.

In the tag identification phase the reader sends out a signal. The tag answers the reader by supplying its current IDS value. On receiving the IDS from the tag, the reader accesses the database and retrieves the ID value and the keys k_1 and k_2 corresponding to the IDS value. The retrieval of these values marks the completion of the tag identification phase.

In the mutual authentication phase the reader generates two random numbers n_1 and n_2 which are employed extensively in generating the messages to be exchanged between the tag and the reader. The protocol uses a function called *MixBits*, which is inherited from the Gossamer protocol. This function is used to generate the values n_3 and $\tilde{n}_1$. The reader now embeds the values of n_1 and n_2 in messages A and B (equations (15) and (16)), in which encryption is done using the IDS value received earlier and the keys k_1 and k_2. The reader also sends a message C (equation (19)) to the tag which is used by the tag to authenticate the reader. The tag extracts the values of n_1 and n_2 from the messages A and B, and then independently calculates C using those values. A comparison between this value and the value of C received from the reader suffices to verify the authenticity of the reader to the tag.

$$n_3 := MixBits\ (n_1, n_2) \quad (13)$$

$$\tilde{n}_1 := MixBits\ (n_3, n_2) \quad (14)$$

$$\mathrm{A} := \mathrm{ROT}((\mathrm{ROT}(\mathrm{IDS} + k_1 + c + n_1,\ k_2) + k_1,\ k_1) \quad (15)$$

$$\mathrm{B} := \mathrm{ROT}((\mathrm{ROT}(\mathrm{IDS} + k_2 + c + n_2,\ k_1) + k_2,\ k_2) \quad (16)$$

$$k_1^* := \mathrm{ROT}((\ \mathrm{ROT}(\ n_2 + k_1 + c + n_3, n_2) + k_2 \oplus, n_1) \oplus n_3 \quad (17)$$

$$k_2^* := \mathrm{ROT}((\mathrm{ROT}(n_1 + k_2 + c + n_3, n_1) + k_1 +\ n_3,\ n_2) + n_3 \quad (18)$$

$$\mathrm{C} := \mathrm{ROT}((\mathrm{ROT}(n_3 + k_1^* + c + \tilde{n}_1, n_3) + k_2^* \oplus \tilde{n}_1, n_2) \oplus \tilde{n}_1 \quad (19)$$

The tag now needs to be authenticated by the reader. For this the tag generates message D and transmits it to the reader. What is significant here is that the tag sends its ID value through message D, to the reader.

In the Gossamer protocol, the value of ID is directly embedded in the message D. Since the operations involved in the generation of message D are rather simple and lightweight, the level of security is compromised, thereby laying the floor open for an attacker who can lay his hands on the all-important ID value. The ID of the tag, which is unique to each tag, can be used thereafter, to impersonate the original tag. The proposed algorithm introduces one-dimensional circular convolution [11] with the Sign/Logarithm number system to encrypt the ID before it can be embedded in message D.

Algorithm ConvolveID
{
1. Split the ID into 12 blocks of size 8-bits each;
2. Divide the random number n_1 into six blocks of size 16-bits each;
3. Convolute the ID blocks using the n_1 blocks as kernel, to give 12 outputs each of 8-bits length;
4. Combine the 12 outputs to form Y which is of 96 bits length;

}

Steps 1 and 2 of the algorithm are depicted in Figures 1 and 2 respectively.

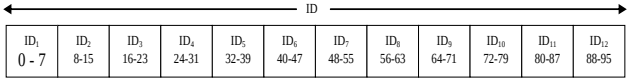

Figure 1. ID value split into 12 blocks

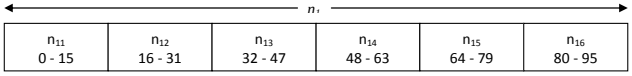

Figure 2. The kernel n_1 split into 6 blocks

Step 3 of Algorithm ConvolveID is carried out using the following equations that represent one dimensional circular convolution. The operations involved are multiplication and addition modulo 2^8.

$$\mathrm{y}_1 := (n_{11} * ID_1) + (n_{12} * ID_2) + (n_{13} * ID_3) + (n_{14} * ID_4) + (n_{15} * ID_5) + (n_{16} * ID_6) \quad (20)$$

$$y_2 := (n_{11} * ID_2) + (n_{12} * ID_3) + (n_{13} * ID_4) + (n_{14} * ID_5) + (n_{15} * ID_6) + (n_{16} * ID_7) \quad (21)$$

$$y_3 := (n_{11} * ID_3) + (n_{12} * ID_4) + (n_{13} * ID_5) + (n_{14} * ID_6) + (n_{15} * ID_7) + (n_{16} * ID_8) \quad (22)$$

$$y_4 := (n_{11} * ID_4) + (n_{12} * ID_5) + (n_{13} * ID_6) + (n_{14} * ID_7) + (n_{15} * ID_8) + (n_{16} * ID_9) \quad (23)$$

$$y_5 := (n_{11} * ID_5) + (n_{12} * ID_6) + (n_{13} * ID_7) + (n_{14} * ID_8) + (n_{15} * ID_9) + (n_{16} * ID_{10}) \quad (24)$$

$$y_6 := (n_{11} * ID_6) + (n_{12} * ID_7) + (n_{13} * ID_8) + (n_{14} * ID_9) + (n_{15} * ID_{10}) + (n_{16} * ID_{11}) \quad (25)$$

$$y_7 := (n_{11} * ID_7) + (n_{12} * ID_8) + (n_{13} * ID_9) + (n_{14} * ID_{10}) + (n_{15} * ID_{11}) + (n_{16} * ID_{12}) \quad (26)$$

$$y_8 := (n_{11} * ID_8) + (n_{12} * ID_9) + (n_{13} * ID_{10}) + (n_{14} * ID_{11}) + (n_{15} * ID_{12}) + (n_{16} * ID_1) \quad (27)$$

$$y_9 := (n_{11} * ID_9) + (n_{12} * ID_{10}) + (n_{13} * ID_{11}) + (n_{14} * ID_{12}) + (n_{15} * ID_1) + (n_{16} * ID_2) \quad (28)$$

$$y_{10} := (n_{11} * ID_{10}) + (n_{12} * ID_{11}) + (n_{13} * ID_{12}) + (n_{14} * ID_1) + (n_{15} * ID_2) + (n_{16} * ID_3) \quad (29)$$

$$y_{11} := (n_{11} * ID_{11}) + (n_{12} * ID_{12}) + (n_{13} * ID_1) + (n_{14} * ID_2) + (n_{15} * ID_3) + (n_{16} * ID_4) \quad (30)$$

$$y_{12} := (n_{11} * ID_{12}) + (n_{12} * ID_1) + (n_{13} * ID_2) + (n_{14} * ID_3) + (n_{15} * ID_4) + (n_{16} * ID_5) \quad (31)$$

$$Y := y_1||y_2||y_3||y_4||y_5||y_6||y_7||y_8||y_9||y_{10}||y_{11}||y_{12} \quad (32)$$

Message Y is formed by concatenating the values of y_1, y_2 ... y_{12}, and is what encrypts the tag's ID that is used in message D. Equation (33) is used to generate message D.

$$D := \mathrm{ROT}(\mathrm{ROT}(n_2 + k_2^* + Y + \tilde{n}_1,\ n_2) + k_2^* + \tilde{n}_1, n_3) + n_1 \quad (33)$$

Partitioning of ID and n_1 need not be just 12 and 6 blocks. This partitioning can be chosen at random and may also be non-uniform. This can be incorporated during fabrication and hence the reader alone, and no attacker, would be privy to the partitioning scheme. Thus, the ID value is secured even further before transmission.

As in the Gossamer protocol, in the IDS and key updating phase, equations (34), (35), (36) and (37) are used by both the reader and the tag to update their own values of IDS, k_1 and k_2. Synchronization between the tag and the reader is ensured by maintaining the values of k_1, k_2 and IDS of the previous iteration.

$$\tilde{n}_2 := \mathrm{MixBits}(\tilde{n}_1, n_3) \quad (34)$$

$$\mathrm{IDS}^{n+1} := \mathrm{ROT}((\mathrm{ROT}(\tilde{n}_1 + k_1^* + \mathrm{IDS}^n + \tilde{n}_2, \tilde{n}_1) + k_2^* \oplus \tilde{n}_2, n_3) \oplus \tilde{n}_2 \quad (35)$$

$$k_1^{n+1} := \mathrm{ROT}((\mathrm{ROT}(n_3 + k_2^* + c + \tilde{n}_2, n_3) + k_1^* + \tilde{n}_2, \tilde{n}_1) + \tilde{n}_2 \quad (36)$$

$$k_2^{n+1} := \mathrm{ROT}((\mathrm{ROT}(\mathrm{IDS}^{n+1} + k_2^* + c + k_1^{n+1}, \mathrm{IDS}^{n+1}) + k_1^* + k_1^{n+1},\ \tilde{n}_2) + k_1^{n+1} \quad (37)$$

where c = 0x3243F6A8885A308D313198A2 (taken from π)

VI. Implementation Specifications

In order for the above protocol to be implemented, the tag needs to maintain a Sign/Logarithm lookup table. It is seen from equations (20) to (31) that the numbers multiplied are of a maximum length of 16-bits. On setting the maximum length of the entries in the lookup table as 16-bits and the step count value as 0.25, it is found that the amount of memory needed in order to store the lookup table works out to less than 1Kbyte. All other operations involved in the generation and decoding of messages also work with 16-bit-long entries of the Sign/Logarithm lookup table.

As the operations other than addition and subtraction are computed using the algorithms of the Sign/Logarithm number system as detailed in Section IV.B, the number of operations involved is a constant irrespective of the values being operated upon. This results in significant power saving.

In general, the choice of precision needed in the lookup table is decided based on the application. With regard to convolution the precision is decided even arbitrarily, and the same should be followed at the reader's side as well. The same lookup table should be maintained at the tag's as well as at the reader's ends. Maintaining the lookup table at the tag's end may be circumvented by ensuring that the required values from the lookup table are sent by the reader to the tag in every run, encrypting such a message with the k_1 and k_2 values that are changed in every run.

The choice of block-size for partitioning ID and n_1 may also be kept open so that the protocol can be custom-tailored to suit the domain's needs.

VII. Conclusion

This newly proposed protocol brings in enhanced security to the Gossamer protocol for lightweight mutual authentication in RFID systems that employ passive RFID tags. Further, it achieves such enhanced security by substantially reducing the power consumption overheads in the existing operations, and by employing the complex operation of one-dimensional circular convolution.

There is usually a trade-off between power consumption and security. The protocol proposed herein, however, achieves a reduction in power consumption using the Sign/Logarithm number system and then introduces complex operations to achieve higher security levels. In fact, with the advent of low power memory, this enhanced protocol can be implemented even in the EPC C1G2 tags. Thus, the protocol proposed in this paper provides an as cheap and yet more secure alternative to the Gossamer protocol, and lends itself for use in large applications with greater security needs.

References

[1] Ari Juels, "*Minimalist cryptography for low-cost RFID tags*", Proceedings of SCN'04, vol. 3352 of LNCS, pages 149–164, 2004.

[2] Association for Automatic Identification and Mobility, "*Radio Frequency Identification RFID – A Basic Primer*", White Paper, Version 1.3, September 2007.

[3] Avery Dennison, "*RFID Basics Updated Including Gen 2*", White Paper, May 2006.

[4] Chien H.Y., "*SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity*", *IEEE Transactions on Dependable and Secure Computing*, vol. 4(4), pp. 337–340. Oct.-Dec. 2007.

[5] Earl E. Swartzlander Jr., and Aristides G. Alexopoulos, "The Sign/Logarithm Number System", IEE Transactions on Computers, Volume 24, Issue 12, December 1975.

[6] Pedro Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags", Handbook of RFIDSec'06, 2006.

[7] Pedro Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "M2AP: A minimalist mutual-authentication protocol for low-cost RFID tags", Proceedings of UIC'06, Vol. 4159 of LNCS, pages 912–923, 2006.

[8] Pedro Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "EMAP: An efficient mutual authentication protocol for low-cost RFID tags". Proceedings of IS'06, Vol. 4277 of LNCS, pages 352–361, 2006.

[9] Pedro Peris-Lopez, "Lightweight Cryptography in Radio Frequency Identification (RFID) systems", Ph.D. Thesis submitted at Universidad Carlos III Demadrid, October 2008.

[10] Pedro Peris-Lopez, Julio Cesar Hernandez-Castro, Juan M. E. Tapiador, and Arturo Ribagorda, "Advances in Ultralightweight Cryptography for Low-cost RFID Tags: Gossamer Protocol", Workshop on Information Security Applications, Vol. 5379 of LNCS, pages 56-68, 2008.

[11] Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing", 2nd Edition, Pearson Eduction, 2003.

[12] Rama N. and Suganya R., "A Family of Gossamer-Based Mutual Authentication Protocols for Tag Pairs in RFID Systems", Reviewed and accepted for presentation and publication in the proceedings of ICMCS International Conference on Mathematics and Computer Science 2010, Loyola College, Chennai.

[13] Rama N. and Suganya R., "Power Analysis of the Gossamer protocol for Passive RFID tags", unpublished, Submitted for review and publication to International Journal of Wireless Communication and Networks on January 2010.

[14] Rama N. And Suganya R., "An Enhanced Power-efficient Gossamer-based Protocol for Passive RFID Tags", unpublished, Submitted for review and publication to International Journal of Computer and Internet Security on February 2010.

[15] Zebra White Paper, "An Introduction to Passive RFID", 2009.

Websites

[16] Class-1 Generation-2 UHF air interface protocol standard version 1.0.9: "Gen2", 2005, url: http://www.epcglobalinc.org/standards/ (last accessed on 06.02.2010).

AUTHORS PROFILE

Dr. Rama N. Completed B.Sc. (Mathematics), Master of Computer Applications and Ph.D. (Computer Science) from the University of Madras, India. She served in faculty positions at Anna Adarsh College, Chennai and as Head of the Department of Computer Science at Bharathi Women's College, Chennai, before moving on to Presidency College, Chennai, where she currently serves as Associate Professor. She has 20 years of teaching experience including 10 years of postgraduate (PG) teaching, and has guided 15 M.Phil. students. She has been the Chairperson of the Board of Studies in Computer Science for UG, and Member, Board of Studies in Computer Science for PG and Research at the University of Madras. Current research interests: Program Security. She is the Member of the Editorial cum Advisory Board of the Oriental Journal of Computer Science and Technology.

Suganya R. Completed B.Sc. (Mathematics) and Master of Computer Applications from the University of Madras, India. She has completed M.Phil. (Computer science) and is currently pursuing Ph.D. (Computer Science) at Mother Teresa Women's University, Kodaikanal, India. She works as Lecturer in Department of Computer Science, Meenakshi College for Women, Chennai, India.