EasySMS: A Protocol for End-to-End Secure Transmission of SMS

Neetesh Saxena, *Member, IEEE,* Narendra S. Chaudhari, *Senior Member, IEEE*

***Abstract*—Nowadays, Short Message Service is being used in many daily life applications including healthcare monitoring, mobile banking, mobile commerce etc. But when we send an SMS from one mobile phone to another, the information contained in the SMS transmit as plain text. Sometimes this information may be confidential like account numbers, passwords, license numbers etc., and it is a major drawback to send such information through SMS while the traditional SMS service does not provide encryption to the information before its transmission. In this paper, we proposed an efficient and secure protocol called EasySMS which provides end-to-end secure communication through SMS between end users. The working of the protocol is presented by considering two different scenarios. The analysis of the proposed protocol shows that this protocol is able to prevent various attacks including SMS disclosure, over the air modification, replay attack, man-in-the-middle attack, and impersonation attack. The EasySMS protocol generates minimum communication and computation overheads as compare to existing SMSSec and PK-SIM protocols. On an average, the EasySMS protocol reduces 51% and 31% of the bandwidth consumption and reduces 62% and 45% of message exchanged during the authentication process as compare to SMSSec and PK-SIM protocols respectively. Authors claim that EasySMS is the first protocol completely based on the symmetric key cryptography and retain original architecture of cellular network.**

***Index Terms*— authentication, over-the-air, security, SMS, symmetric key**

I. Introduction

NOWADAYS Short Message Service (SMS) has become one of the fastest and strong communication channels to transmit the information across the worldwide. On December 3, 2013, SMS service has completed its 21 years as on December 3, 1992, the world's first SMS was sent by Neil Papworth from the UK through the Vodafone network [1]. The SMS are used in many real world applications as a communication medium such as in Transportation Information System [2], MobileDeck [3], SMSAssassin [4], SMS-based web search such as SMSFind [5], Monitoring Community Health Worker Performance [6], private health facilities using SMS [7], participation in elections through SMS [8], in Crime Scene Investigation [9] and many more.

A. *Research Problem*

Sometimes, we send the confidential information like password, pass code, banking details and private identity to our friends, family members and service providers through an SMS. But the traditional SMS service offered by various mobile operators surprisingly does not provide information security of the message being sent over the network. In order to protect such confidential information, it is strongly required to provide end-to-end secure communication between end users. SMS usage is threatened with security concerns, such as SMS disclosure [10], man-in-the-middle attack [11], replay attack [12] and impersonation attack [13]. There are some more issues related to the open functionality of SMS which can incapacitate all voice communications in a metropolitan area [14], and SMS-based mobile botelnet [15] as Android botnet [16]. SMS messages are transmitted as plaintext between mobile user (MS) and the SMS center (SMSC), using wireless network. SMS contents are stored in the systems of network operators and can be read by their personnel.

B. *Key Contribution*

The above requirements can be accomplished by proposing a protocol called EasySMS which provides end-to-end security during the transmission of SMS over the network. The EasySMS protocol prevents the SMS information from various attacks including SMS disclosure, over the air (OTA) modification, replay attack, man-in-the-middle attack, and impersonation attack. This EasySMS sends lesser number of transmitted bits, generates less computation overhead, and reduces bandwidth consumption and message exchanged as compare to SMSSec [17] and PK-SIM [18] protocols.

C. *Organization*

This paper has organized into VII sections. Section II presents literature review of the work done related to SMS security. In section III, a new protocol is proposed which provides end-to-end secure transmission of SMS in cellular networks. Section IV illustrates the analysis of proposed protocol. Section V, discusses suitable symmetric algorithm for EasySMS protocol. Section VI presents formal proof of EasySMS protocol. Finally, section VII summarizes conclusion of the work.

This paragraph of the first footnote will contain the date on which you submitted your paper for review. "This work was supported by TCS India".

N. Saxena is with the Discipline of Computer Science & Engineering, IIT Indore, MP 453441, India, (e-mail: neetesh.saxena@gmail.com).

N. S. Chaudhari is Director and Professor (Computer Science & Engineering), VNIT Nagpur, India, and Professor, Discipline of Computer Science & Engineering, IIT Indore, India, (e-mail: nsc183@gmail.com).

TABLE I
ABBREVIATION AND SYMBOLS

Symbol	Definition	Bits
MS	Mobile Station referring user	_
AS	Authentication Server referring AuC	_
CA/RA	Certification/Registration Authority	_
IDMS	International Mobile Subscriber Identity of MS	128
Q/Qn	New Session Identifier	28
Rc/Nc/Ns/Na	Random Number	128
Pf	Private Port Number	16
ReqNo	Request Number	8
SK/SK_MS	Symmetric key shared b/w MS and AS	128
DK1	Delegation key	256
MAC/H	Message Authentication Code/Hash	64
Ti	Timestamp	64
CertSAG	Certificate of Security Access Gateway	40
SK_AS-CA	Symmetric key shared b/w AS and CA/RA	128
SK_AS1-AS2	Symmetric key shared b/w AS1 and AS2	128
SQ/Seq	Sequence Number	28
PK/PK_PK-SIM	Public key of Server	128
UAKey	Primary key	128
Expiry/ExpT	Expiry Time	64

TABLE II
DEFINITION OF FUNCTIONS USED

Symbol	Definition
f1	Message authentication code function
f2	Key generation function for DK1
{}SK_AS-CA	Encryption function with SK_AS-CA key
{}SK_AS1-AS2	Encryption function with SK_AS1-AS2 key
{}DK1	Encryption function with DK1 key
{}SK_MS2	Function with symmetric key of MS2 shared b/w MS2 and AS/AS2
||	Concatenation

II. RELATED WORK

Previously, various authors have proposed different techniques to provide security to the transmitted messages. An implementation of a public key cryptosystem for SMS in a mobile phone network has been presented in [19] but the security analysis of the protocol has not discussed. A secure SMS is considered to provide mobile commerce services in [20] and is based on public key infrastructure. A framework Secure Extensible and Efficient SMS (SEESMS) is presented in [21] which allows two peers to exchange encrypted communication between peers by using public key cryptography. Another new application layer framework called SSMS is introduced in [22] to efficiently embed the desired security attributes in SMS to be used as a secure bearer for m-payment systems and solution is based on the elliptic curve-based public key that uses public keys for the secret key establishment. An efficient framework for automated acquisition and storage of medical data using the SMS based infrastructure is presented in [23] and the results conclude that the proposed SMS based framework provides a low-bandwidth, reliable, efficient and cost effective solution for medical data acquisition. The [20] and [22] generate shared key for each session but also generate huge overheads and not suitable for the real world applications. In all [19], [20], [21], [22] and [23], it is not clear whether the proposed approaches are able to prevent SMS against various attacks. All the above mentioned approaches/ protocols/ frameworks generate a large overhead as they propose an additional framework for the security of SMS. Due to physical limitations of the mobile phones, it is recommended to develop a protocol which would make minimum use of computing resources and would provide better security. However, implementation of framework always increases the overall overhead which is not much suitable for the resource constraints devices such as mobile phones. Thus, in this paper we compared our proposed protocol with the existing SMSSec and PK-SIM protocols.

The reason for chosen these protocols for comparison is that these are the only existing protocols which do not propose to change the existing architecture of cellular networks. We wanted to compare our proposed protocol with some existing protocols devoted to provide end-to-end SMS security with symmetric key cryptography, but there is no such protocol exists. Both protocols are having two phases similar to the proposed protocol and are based on symmetric as well as asymmetric key cryptography while the proposed protocol is completely based on symmetric key cryptography. The SMSSec protocol can be used to secure an SMS communication sent by Java’s Wireless Messaging API while the PK-SIM protocol proposes a standard SIM card with additional PKI functionality. Both protocols are based on client-server paradigm, i.e., one side is mobile user and the other side is authentication server but they do not present any scenario where an SMS is sent from one mobile user to another mobile user. The SMSSec protocol does not illustrate the security analysis.

III. SECURITY GOALS & PROPOSED SOLUTION

This section focuses on the attack model, system and communication model, basic assumption and detail description of proposed protocol. Table I represents definition of various symbols used in the paper with their sizes, while Table II lists various functions used in the paper with their definitions.

A. Attack Model

An attack model describes different scenarios for the possibilities of various attacks where a malicious MS can access the authentic information, or misguide the legitimate MS. Since, the SMS is sent as plaintext, thus network operators can easily access the content of SMS during the transmission at SMSC. This leads to SMS disclosure attack. In traditional cellular network, the OTA interface between the MS and the Base Transceiver Station (BTS) is protected by a weak encryption algorithm (such as A5/1 or A5/2), thus an attacker can compromise these algorithms to capture the information contained in the SMS or can alter the SMS information. The attacker can also try to cryptanalyze the generated cryptographic keys used in the authentication protocol. The attacker may fraudulently delay the conversation between both MS and can capture or reuse the authenticated information (during the protocol execution) contain in previous messages which results in the form of replay attack. Later, the attacker may send the captured information to the server or can modify the sequence of messages for getting the authentication token. An attacker can also perform a man-in-the-middle attack when an MS is connected to a BTS through wireless network and eavesdrops the session initiated by legitimate MS. The attacker establishes an independent connection with both the victim's MS. It performs eavesdropping on the active connection, modifies and intercepts the messages. However, the intruder must intercept the transmitted message between two victim MS and inject false information, which is straightforward in the circumstances where communication is done in an unencrypted or weak encryption network. But all is possible when an attacker gets the secret key or some information based on which he/she could guess the secret key. Normally, this attack executes during the key exchange phase of the protocol and tries to capture the session key. It may happen that the intruder could impersonate the MS or the AS, if the proper integrity is not maintained over the network. The intruder can pretend like a legitimate MS and ask to the AS for valid authentication tokens in order to make the AS believe that originate from the authentic MS. Similarly, he/she can also show him(her)self like a valid AS and ask legitimate MS to send the information in order to make the target MS believe that originate from a genuine AS.

B. System and Communication Model

In order to overcome the above stated attacks, various cipher algorithms are implemented with the proposed authentication protocol. We recommend that the cipher algorithms should be stored on to the SIM (part of MS) as well as in the AS. Since providing security needs to do some extra effort which is measured in terms of cost, thus providing or adding extra security means increasing more cost. Authors propose to include one more service as 'Secure Message' in the menu of mobile software developed by various mobile companies as shown in Fig. 1. Mobile operators can add some extra charges to send secure message by their customers over the networks. Whenever a user wants to send a secure message to other user, the proposed protocol namely EasySMS is executed which makes available the symmetric shared key between both MS and then ciphering of message takes place using a symmetric key algorithm.

C. *Proposed Protocol: EasySMS*

In this section, we propose a new protocol named EasySMS with two different scenarios which provide end-to-end secure transmission of information in the cellular networks. First scenario is illustrated in Fig. 2 where both MS belong to the same AS, in other words share the same Home Location Register (HLR) while the second scenario is presented in Fig. 3 where both MS belong to different AS, in other words both are in different HLR. There are two main entities in the EasySMS protocol. First is the Authentication Server (AS), works as Authentication Center (AuC) and stores all the symmetric keys shared between AS and the respective MS. In this paper, we refer AuC as the AS. Second entity is the Certified Authority/ Registration Authority (CA/RA) which stores all the information related to the mobile subscribers. We assume that every subscriber has to register his/her mobile number with CA/RA entity and only after the verification of identity, the SIM card gets activated by this entity. Thus, this entity is responsible to validate the identity of the subscribers.

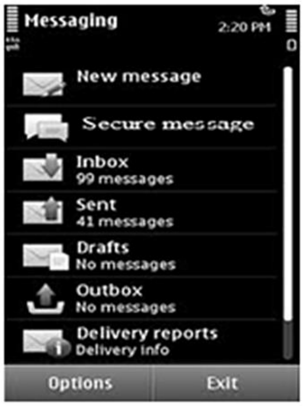

Fig. 1. Secure Message in Menu

We also assume that a symmetric key is shared between the AS and the CA/RA which provides the proper security to all the transmitted information between AS and CA/RA. It is considered that various authentication servers are connected with each other through a secure channel since one centralized server is not efficient to handle data all around. We consider all the transmission among various AS take place by encrypting the message with a symmetric key shared between each pair of AS. Both scenarios of this protocol are as follows:

Scenario-1: When both MS belong to same AS: This scenario is presented in Fig. 2 where MS1 sends a message to MS2 and both MS belong to the same AS. This scenario is subdivided into two phases.

Phase-1: (1) First, the mobile user who wants to send the SMS (say MS1) transmits an initial request to other mobile user (say MS2) for the connection. This initial request consists of International Mobile Subscriber Identity (IMSI) of MS1 (say IDMS1), a timestamp T1, a request number ReqNo and a message authentication code MAC1 = f1SK1(IDMS1 ||ReqNo). Here, SK1 is a symmetric key shared between the MS1 and the AS2. (2) On receiving the message from MS1, the mobile user who receives this request (say MS2) computes the MAC2 = f1SK2 (IDMS2||T2||MAC1). Then MS2 sends a message to the AS containing the IDMS1, IDMS2, T2, MAC1, ReqNo and MAC2 where IDMS2 is the IMSI of the MS2. The SK2 is a symmetric key shared between MS2 and the AS.

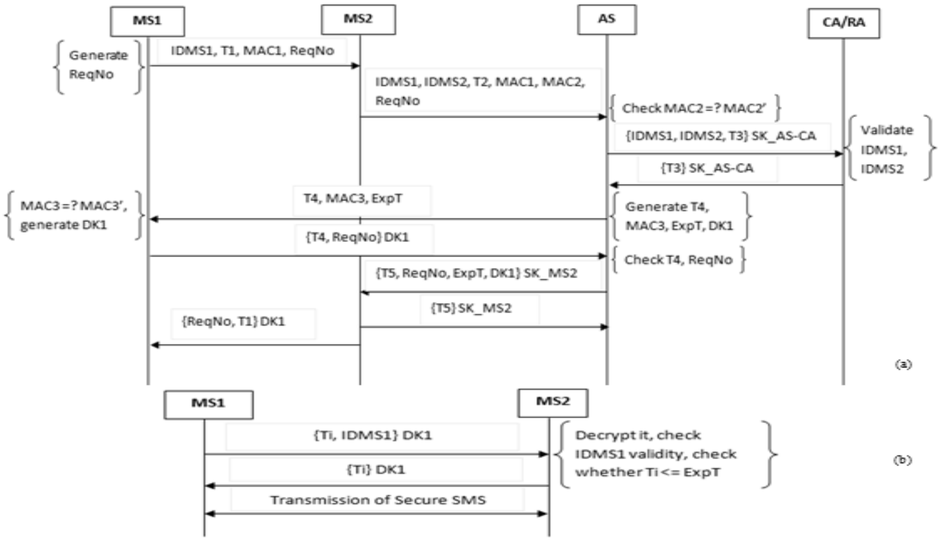
MS1
MS2
AS
CA/RA
Generate ReqNo
IDMS1, T1, MAC1, ReqNo
IDMS1, IDMS2, T2, MAC1, MAC2, ReqNo
Check MAC2 =? MAC2'
{IDMS1, IDMS2, T3} SK_AS-CA
Validate IDMS1, IDMS2
{T3} SK_AS-CA
MAC3 =? MAC3', generate DK1
T4, MAC3, ExpT
Generate T4, MAC3, ExpT, DK1
{T4, ReqNo} DK1
Check T4, ReqNo
{T5, ReqNo, ExpT, DK1} SK_MS2
{T5} SK_MS2
{ReqNo, T1} DK1
(a)
MS1
MS2
{Ti, IDMS1} DK1
Decrypt it, check IDMS1 validity, check whether Ti <= ExpT
{Ti} DK1
Transmission of Secure SMS
(b)

Fig. 2. EasySMS Scenario 1: (a) Phase-1 (b) Phase-2

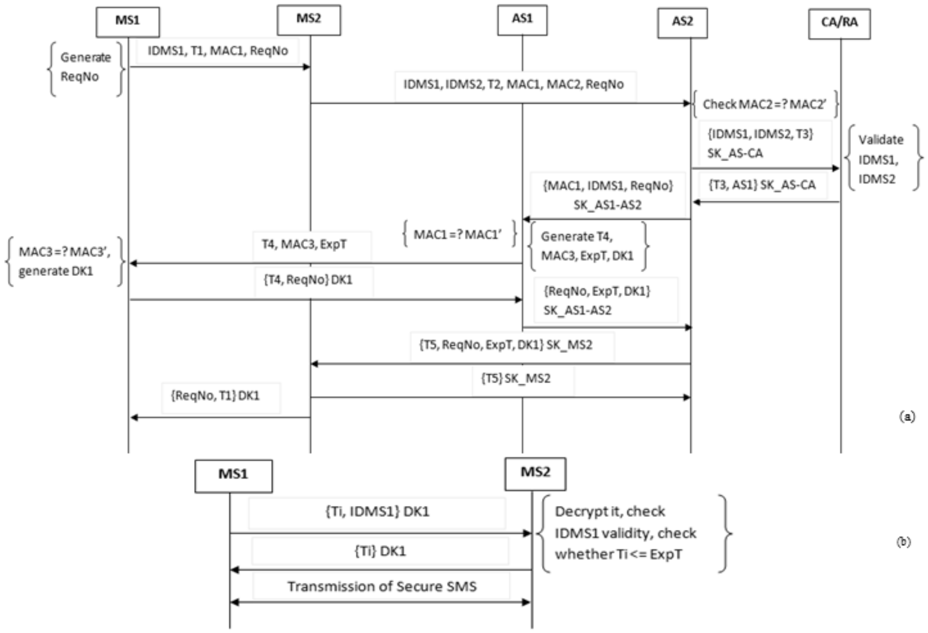
MS1
MS2
AS1
AS2
CA/RA
Generate ReqNo
IDMS1, T1, MAC1, ReqNo
IDMS1, IDMS2, T2, MAC1, MAC2, ReqNo
Check MAC2 =? MAC2'
{IDMS1, IDMS2, T3} SK_AS-CA
Validate IDMS1, IDMS2
{MAC1, IDMS1, ReqNo} SK_AS1-AS2
{T3, AS1} SK_AS-CA
MAC1 =? MAC1'
Generate T4, MAC3, ExpT, DK1
MAC3 =? MAC3', generate DK1
T4, MAC3, ExpT
{T4, ReqNo} DK1
{ReqNo, ExpT, DK1} SK_AS1-AS2
{T5, ReqNo, ExpT, DK1} SK_MS2
{T5} SK_MS2
{ReqNo, T1} DK1
(a)
MS1
MS2
{Ti, IDMS1} DK1
Decrypt it, check IDMS1 validity, check whether Ti <= ExpT
{Ti} DK1
Transmission of Secure SMS
(b)

Fig. 3. EasySMS Scenario 2: (a) Phase-1 (b) Phase-2

With this message, the MS2 requests to the AS to check the validity of the IDMS1. (3) When the AS receives a message from the MS2, it computes the MAC2' = f1SK2(IDMS2||T2||MAC1) and compares it with the received MAC2. If it holds then the AS sends not only the IDMS1 but also the IDMS2 to the CA/RA along with a timestamp T3 using a symmetric shared key between AS and CA/RA (say SK_AS-CA) to validate the identity of both MS. If, MAC2 and MAC2' are not equal then the connection is terminated. (4) Next, the CA/RA checks the validity of both entities and sends the reply back to the AS with the received timestamp T3. (5) On receiving the message from the CA/RA, if the AS finds any of the entities is invalid then the connection is simply terminated and MS1 needs to send a fresh connection request. If both entities are valid then the AS generates a new timestamp T4, an expiry time to authenticate MS1 (say ExpT), a delegate key DK1 generated from the SK1 using a function f2 and a new message authentication code MAC3=f1SK1(T4|| ExpT||ReqNo) and DK1= f2SK1(T4||ReqNo). Then the AS sends (T4, MAC3, ExpT) to the MS1. (6) After receiving the message from AS, the MS1 first computes MAC3' and compares it with the received MAC3, where MAC3'= f1SK1(T4||ExpT||ReqNo). If both are same then MS1 computes the DK1. Next, MS1 sends T4 and the corresponding ReqNo to the AS encrypted with the DK1 key. (7) The AS checks the received T4 with its stored value and confirms ReqNo. If both are correct then the authentication of MS1 is completed and then after AS sends DK1 to the MS2 along with a new timestamp T5, ExpT and ReqNo after encrypting all using the SK of MS2 (SK_MS2) which is a shared key between AS and MS2. (8) The MS2 simply confirms the reception of DK1 key by replying to the AS, the

T5 encrypted with the SK of MS2. (9) MS2 also sends ReqNo and T1 to the MS1 encrypted with DK1 so that MS1 can verify the correctness of T1 and ReqNo. This message also verifies the successful reception of DK1 by the MS2.

Phase-2: Once both MS have a shared secret symmetric key, they can exchange the message information in a secure manner using a suitable and strong cryptographic algorithm like AES/ MAES (explained later). After phase-1, a session is generated which provides the secure communication between both MS for a specified time period ExpT. In this time period the same DK1 key is used to provide ciphering between MS1 and MS2 but after the ExpT time the session gets expire and MS1 needs to send a fresh request to MS2 with a new request number ReqNo with the same procedure of phase-1. Within the ExpT, the following steps are used for the communication between both MS: (1) The MS1 sends the IDMS1 and a timestamp (say Ti) to the MS2 encrypted with symmetric key of MS1 i.e., DK1. (2) MS2 decrypts the message using the same DK1 key and checks the validity of IDMS1 and verifies whether Ti <= ExpT. If both are correct then MS1 is successfully authenticated and proved as a valid user for the connection. Then MS2 replies the same received Ti encrypted with DK1 as an acknowledgement to MS1. (3) Secure SMS communication between both MS takes place.

Scenario-2: When both MS belong to different AS: This scenario is presented in Fig. 3 where MS1 sends a message to MS2 while both MS belong to the different AS. This case is one where both mobile users are located in the geographically far areas and they have different authentication centers. It may be the case where both MS are of different service providers so they genuinely have different authentication centers. This scenario is also subdivided into two phases.

Phase-1: (1) It is same as presented in step-1 of scenario-1. Here, SK1 is a symmetric key shared between MS1 and AS1. (2) The MS2 passes (IDMS1, IDMS2, ReqNo, T2, MAC1, MAC2) to the AS through which it is connected (say AS2). The SK2 is a symmetric key shared between M[illegible] (b) the AS2. With this message, the MS2 requests to the AS2 to check the validity of the IDMS1. The MS2 stores the timestamp T1 in the memory which was received from the MS1. (3) The AS2 computes the same as presented in step-3 of scenario-1 and checks whether MAC2?=MAC2'. (4) The CA/RA checks the validity of both entities and sends the reply back to the AS2 with the received timestamp T3 and the identity of AS to which MS1 belongs (say AS1). (5) The AS2 checks th[illegible] (a) as in scenario-1 step-5, if both entities are valid then th[illegible] sends (IDMS1, ReqNo, MAC1) to the AS1 through a secure channel or using a symmetric key shared between AS1 and AS2 (say SK_AS1-AS2). We assume that all AS communicate with each other using the pre-computed symmetric shard keys. (6) When the AS1 receives the message from the AS2, it computes MAC1'= f1SK1(IDMS1||ReqNo) and compares MAC1' with the received MAC1. If both are different then the connection is terminated. If both are same then the AS1 generates a new timestamp T4, an expiry time to authenticate MS1 (say ExpT), a delegate key DK1 generated from the SK1 of MS1 using a function f2, and a MAC3, where MAC3 = f1SK1(T4||ExpT|| ReqNo) and DK1 = f2SK1(T4|| ReqNo). Then the AS1 sends (T4, MAC3, ExpT) to the MS1. (7) After receiving the message from AS1, MS1 repeats the same as in scenario-1 step-6 and sends (T4, ReqNo) to the AS1 encrypted with DK1 key. (8) The AS1 checks T4 and ReqNo as in scenario-1 step-7. Then AS1 conveys the confirmation of the authentication of MS1 by sending a message (ReqNo, ExpT, DK1) to the AS2 using SK_AS1-AS2 key. (9) The AS2 sends DK1 to the MS2 along with a new timestamp T5, expiry time ExpT and request number ReqNo after encrypting all using the SK of MS2 (say SK_MS2) which is a shared key between the AS2 and the MS2. (10) MS2 repeats the same as in scenario-1 step-8, and sends encrypted reply of T5 to the AS2. (11) It is same as in scenario-1 step-9.

Phase-2: The phase-2 is same as discussed in the previous scenario of phase-2.

IV. Analysis of Proposed Protocol

This section analyzes proposed protocol in various aspects such as mutual authentication, prevention from various threats and attacks, key management, and computation & communication overheads.

Is the Secret Key SK Safely Stored? Since any malicious user does not know the structure of cryptographic functions like f1() and f2(), so he/she can neither generate the correct MAC1 nor correct delegation key DK1. The secret key SK is stored in the authentication server/center as well as embedded onto the SIM at the time of manufacturing. Thus, it is almost impossible to extract the SK. The storage scenario of SK key we presented is same as nowadays used for the voice communication in the traditional cellular networks. If some service providers do not wish to use actual SK in the protocol execution, they can compute alternate secret keys with a new function f'' as: SK1' =f''SK1(IDMS1) and SK2'= f''SK2(IDMS2). We do not prefer to do it because it increases the overall overhead of protocol.

Is There Any Alternative for IMSI? Since a malicious user with only known IMSI (by some IMSI catcher but functions and secret keys are still unknown) cannot break the security of proposed protocol. Thus, the proposed protocol is secure. We can also have one alternate for it. We can propose a new function f'() which computes a temporary IMSI for each MS whenever it wants to communicate. At MS: compute IDMS1 = f'(IMSI1, MAC1); At AS: compute IMSI1 = f'(IDMS1, MAC1). This is simply possible by XORing the IMSI1 (or IDMS1) and MAC1 (twice), because the size of MAC1 is 64 bits while IMSI1/IDMS1 is of 128 bits. The function f'() should be known to MS as well as AS but publically unknown. But we recommend using a complex function to compute the same. However, we do not prefer because it increases the overhead at MS as well as at AS.

A. Mutual Authentication between MS and AS

In scenario-1 of EasySMS protocol, the AS authenticates MS1 by verifying the MAC2 and checks the identity of MS1 through CA/RA. When AS receives MAC2, it simply calculates MAC2' and compares it with the received MAC2. If it matches, then authentication of MS1 is done by the AS.

TABLE IV
MESSAGE EXCHANGED RATIO

No.of Auth. Requests	EasySMS/SMSSec	EasySMS/PK-SIM
10	0.55	0.75
50	0.38	0.55
100	0.35	0.
200	0.34	0.62
500	0.33	0.61
1000	0.33	0.6
Average	**0.38**	**0.69**

Similarly, on receiving MAC3, the MS1 computes MAC3' to authenticate the AS. If MAC3 is equal to the MAC3' then the authentication of AS is successful. This all ensures the mutual authentication between MS1 and AS through MS2. Similarly, in scenario-2, the AS1 authenticates MS1 through AS2 and MS2. The integrity is maintained between MS1-AS1 and MS2-AS2 by comparing the MAC1-MAC1' and MAC2-MAC2' respectively. The MS1 authenticates AS1 by comparing MAC3 with MAC3'.

B. Efficient Key Management

The EasySMS protocol is able to efficiently handle the key management issue in both scenarios where the DK1 key (from the symmetric key of MS1) is securely transmitted by the AS to the MS2 (scenario-1) or by the AS2 to the MS2 through AS1 (scenario-2). Thus, this protocol successfully ciphers the message before its transmission over the network. We preferred a symmetric key algorithm because these algorithms are 1000 times faster than the asymmetric algorithms [24] and improve the efficiency of the system.

C. Resistance to Attacks

In this subsection, we justify that the EasySMS protocol is able to prevent the transmitted SMS from various attacks over the network. It is assumed that the cryptographic functions used in the paper are not publically available and are secret. The capturing of any secret key SK is not possible because no secret key has been transmitted in any phase of the proposed protocol and always a delegation key DK1 is being transferred in cipher form whenever is required. Secret keys are also not publically available and are secret.

1) SMS Disclosure: In EasySMS protocol, a cryptographic encryption algorithm AES/MAES is maintained to provide end-to-end confidentiality to the transmitted SMS in the network. Thus, encryption approach prevents the transmitted SMS from SMS disclosure.

2) Replay Attack: The proposed protocol is free from this attack because it sends one timestamp (like T1, T2, T3, T4 and T5) with each message during the communication over the network. These unique timestamp values prevent the system from the replay attack. This attack can be detected if later previous information is used or modified.

3) Man-in-the-Middle Attack: In EasySMS protocol, a symmetric algorithm AES/MAES is used for encrypting/ decrypting end-to-end communication between the MS and the AS in both scenarios. The message is end-to-end securely encrypted/decrypted with DK1 key for every subsequent authentication and since attacker does not have sufficient information to generate DK1, thus it prevents the communication from MITM attack over the network.

4) OTA Modification in SMS Transmission: The EasySMS protocol provides end-to-end security to the SMS from the sender to the receiver including OTA interface with an additional strong encryption algorithm AES/MAES. The protocol does not depend upon the cryptographic security of encryption algorithm (such as A5/1, A5/2) exists between MS and BTS in traditional cellular networks. This protocol provides end-to-end security to end users. It protects the message content being access by mobile operators as well as from attackers present in the transmitted medium.

5) Impersonation Attack: There are two cases to evaluate this attack with EasySMS protocol. Both cases are as follows: (a) *When an attacker impersonates the MS:* In EasySMS protocol, if an attacker tries to impersonate the MS, he/she will not get success because in scenario-1, the AS calculates the MAC2' and compares it with the received MAC2 while in scenario-2, the AS2 computes MAC2' and compares with MAC2 and after that AS1 computes MAC1' and checks whether MAC1' is equal to the MAC1. Thus, at any stage if the AS finds the above comparison false then the connection is simply terminated. (b) *When an attacker impersonates the AS:* If an attacker tries to impersonate the AS (or AS1/AS2), the attempt to impersonate the AS will be failed as the MS1 computes MAC3' and compares it with the received MAC3. Thus, an attempt to impersonate the AS terminates the connection.

D. Computation Overhead

We have considered all the security functions used in EasySMS, SMSSec, and PK-SIM a unit value. On the basis of authentication requests 'n' and number of functions used in three protocols, we calculate computation overhead as:

1) SMSSec Protocol: Phase-1: [H, {}PK, {}SK, {}SK, {}SK] = 5; *Phase-2:* [H, HU, {}SK, {}SK_n, {}SK_n, {}SK_n]*n = 6*n; Total Overhead = 5+6*n

2) PK-SIM Protocol: Phase-1: [H(CertSAG), {}SK_SAG, H(C_ME), {}SK_SAG, H(Ns, Nc, UAKey, Expiry), {}SK_SAG, {}PK_PK-SIM, {}E_UAKey]=8; *Phase-2:* [MAC, {}E_SK, MAC', {}E_SK]*n = 4*n; Total = 8+4*n

TABLE III
BANDWIDTH UTILIZATION

No.of Auth. Requests	EasySMS/SMSSec	EasySMS/PK-SIM
10	0.76	0.99
50	0.48	0.69
100	0.45	0.64
200	0.43	0.62
500	0.42	0.61
1000	0.41	0.6
Average	**0.49**	**0.69**

3) EasySMS Protocol: Scenario-1: Phase-1: f1, f1, f1, f1, f1, f2, f2, {}SK_AS-CA, {}SK_AS-CA, {}SK_MS2, {}SK_MS2, {}DK1, {}DK1 = 13; *Phase-2:* [{}DK1, {}DK1]*n = 2*n; Total Computation Overhead = 13+2*n

Scenario-2: Phase-1: f1, f1, f1, f1, f1, f1, f2, f2, {}SK_AS-CA, {}SK_AS-CA, {}SK_AS1-AS2, {}SK_AS1-AS2, {}SK_MS2, {}SK_MS2, {}DK1, {}DK1 = 16; *Phase-2:* [{}DK1, {}DK1]*n = 2*n; Total Overhead = 16+2*n

E. Communication Overhead

In this subsection, we calculate the transmitted message size to evaluate communication overhead in EasySMS, SMSSec, and PK-SIM protocols. The total number of transmitted bits can be calculated with the help of the size specified in Table I. Total number of transmitted bits in each protocol is as:

1) SMSSec Protocol: Phase-1: (1)+(2)+(3)+(4) = (40+64+64 +28+128)+(128+16+28)+(28)+(28) = 552 bits; *Phase-2:* (for n values) = ((1)+(2)+(3)+(4))*n = ((64+40+64+64+28+128) +(128+16+28)+(28)+(28))*n = 616*n; Total bits = 552 + 616*n; Here, random number Rc is 128 bits.

2) PK-SIM Protocol: Phase-1:(1)+(2)+(3)+(4)+(5)=(40+128 +64+28)+(40)+(40+64)+(128+128+64+64)+(128) = 916 bits; *Phase-2:* (for n values) = ((1)+(2))*n = ((40+128+64)+(128 + 64))*n = 424*n; Total transmitted bits = 916 + 424*n

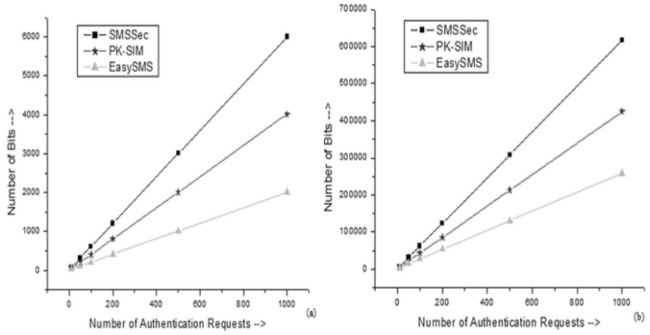

Fig. 4. (a) Computation (b) Communication Overhead

3) EasySMS Protocol: Case-1: Phase-1: (1)+(2)+ (3)+(4)+ (5)+(6)+(7)+(8)+(9)=(128+64+64+8)+(128+128+64+64+64+ 8)+(128+128+64)+(64)+(64+64+64)+(64+8)+(64+8+64+256) +(64)+(64+8) = 1896 bits; *Phase-2:* ((1)+(2))*n = ((64+ 128)+(64))*n = 256*n bits; Total bits = 1896 + 256*n bits

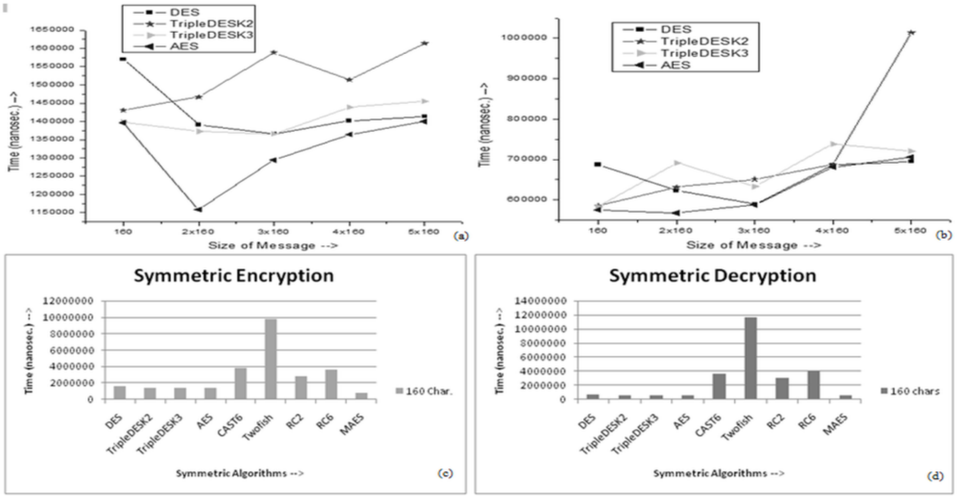

Fig. 5. Encryption and Decryption with different size of messages

Case-2: Consider the identity of AS1 is 128 bits. *Phase-1:* (1)+(2)+(3)+(4)+(5)+(6)+(7)+(8)+(9)+(10)+(11)=(128+64+64 +8)+(128+128+64+64+64+8)+(128+128+64)+(64+128)+(64+128+8)+(64+64+64)+(64+8)+(8+64+256)+(64+8+64+256)+ (64)+(8+64)=2552 bits; *Phase-2:* ((1))+(2))*n = ((64+ 128)+(64)*n= 256*n bits; Total bits = 2552 + 256*n

Fig. 4 shows the graphs between the number of bits and the number of authentication requests generated. It can be clearly seen that EasySMS generates lesser computation overhead (Fig. 4(a)) and communication overhead (Fig. 4(b)) as compare to SMSSec and PK-SIM protocols.

F. Bandwidth Utilization

This subsection evaluates the bandwidth utilized by all three protocols and compares them with respect to each other. Table III presents the bandwidth utilization of EasySMS with respect to SMSSec and PK-SIM protocols. It can be easily concluded that on an average, the EasySMS protocol reduces 51% and 31% of the bandwidth consumption during the authentication process as compare to SMSSec and PK-SIM respectively, while the number of authentication requests is considered as 10, 50, 100, 200, 500, 1000. Similarly, Table IV shows that proposed protocol reduces 62% and 45% of the message exchanged in comparison both protocols respectively.

V. SYMMETRIC ENCRYPTION ALGORITHM

In this section, we focus on the selection criteria to choose a block cipher based symmetric key algorithm. The performance of a block cipher varies with the block size and key size. Larger the block size, faster will be the algorithm, because with a larger block size, a large chunk of the data will be encrypted in a single execution cycle of the algorithm.

TABLE V
MESSAGE SIZE (PLAIN TEXT, CIPHER TEXT)

Mode	DES	TripleDES-2K	TripleDES-3K	AES
PCBC	160, 80	160, 155	160, 155	160, 80
ECB	160, 143	160, 160	160, 168	160, 80
CBC	160, 80	160, 156	160, 156	160, 155
CTR	160, 82	160, 82	160, 82	160, 160
CFB	160, 82	160, 82	160, 159	160, 161
OFB	160, 161	160, 82	160, 82	160, 82

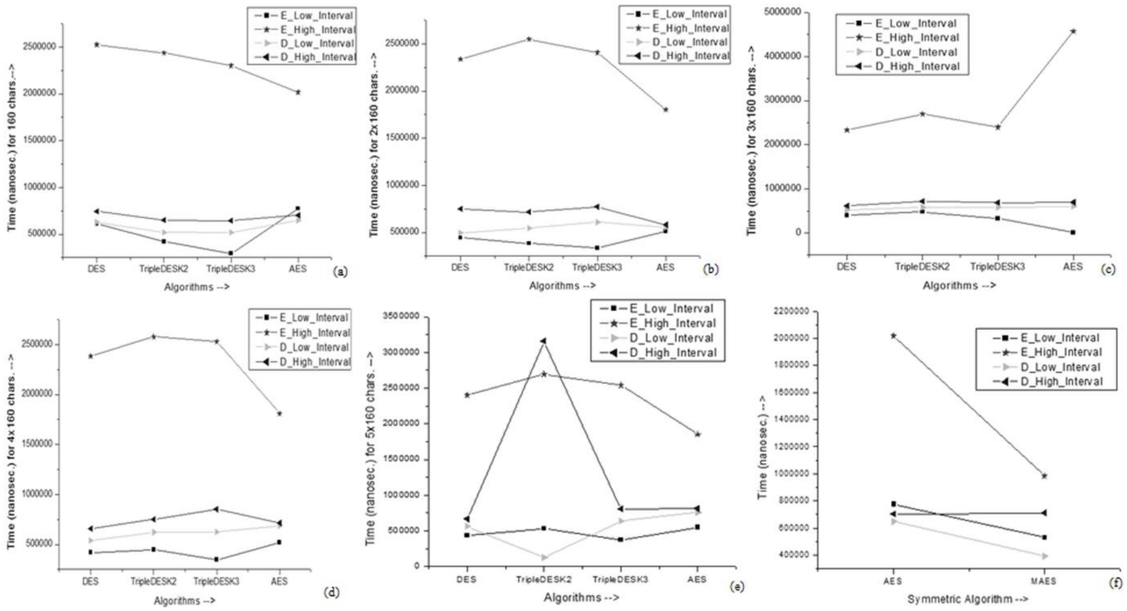

Fig. 6. Confidence interval with SMS Size (char.) (a) 160 (b) 2x160 (c) 3x160 (d) 4x160 (e) 5x160 (f) 160

Likewise, a larger key results in a slower algorithm, because in general, all bits of the key are involved in an execution cycle of the algorithm. A large number of rounds make the algorithm slower but are supposed to provide greater security [25]. Thus, there is always a trade-off between security and performance in block cipher algorithms [26]. Eli Biham [27] has suggested that performance of algorithm should be measured by timing the minimum number of secure rounds for each algorithm, i.e., the estimated number of rounds needed to make a brute force key search which is the most efficient form of attack but there is no easy way of obtaining impartial and widely accepted values for the minimum number of secure rounds for each algorithm. In J2ME, the WMA (Wireless Messaging API) [28] provides tools for sending and receiving SMS messages. Our solution is based on JDK 1.6 and is simulated with Java MIDlet, which is an application written in Java for the Micro Edition platform. The application can send and receive SMS messages in binary format using the WMA. Since the J2ME does not contain cryptographic algorithms, we used Lightweight API from the Legion of the Bouncy Castle.

A. Simulation

Some existing symmetric key algorithms like DES, Triple-DES with 2-keys, Triple-DES with 3-keys, and AES have been implemented. The results have generated on a PC with configuration of Core i3 processor, 4 GB RAM, 320 GB HD and Windows7 OS. J2ME implementation of these algorithms is limited with 160 characters only, i.e., single SMS. We have used JDK 1.6 for the implementation of these algorithms with more than 160 characters. The standard key size used in DES, Triple DES with 2-keys, Triple-DES with 3-keys and AES are 64 (out of which 56 bits are used), 112, 168, and 128 bits respectively. Fig. 5(a) and Fig. 5(b) show the results observed through JDK1.6 for encryption and decryption with DES, Triple-DES with 2-keys, Triple-DES with 3-keys, and AES. The results conclude that out of these algorithms, AES takes minimum time to encrypt and decrypt the SMS with various sizes where one SMS size is 160 characters. Table V represents the pairs of plain text and cipher text with respect to various algorithms DES, AES, Triple-DES with 2-keys, and Triple-DES with 3-keys implemented in various modes of operations like Propagation Chain Block Cipher (PCBC), Electronic Code Book (ECB), Chain Block Cipher (CBC), Counter (CTR), Output Feedback Block (OFB) and Cipher Feedback Block (CFB). Out of all these modes, CTR mode is the most popular and usable, because it provides the parallelism to encrypt and decrypt all blocks of data simultaneously. Nowadays, DES and Triple-DES algorithms are not considered as very secure algorithms [29], [30] since previously some attacks have been found on both algorithms. Thus, AES is the best option for this purpose which is considered one of the best secure algorithms. With the input of 160 characters, DES, AES, Triple-DES with 2-keys, and Triple-DES with 3-keys algorithms in CTR mode generate 82, 82, 82 and 160 characters cipher respectively, which means through AES, we can still send 160 characters after encrypting the SMS. Each algorithm results are calculated 30 times by repeating execution and the average value is considered.

B. Reliability Analysis with Confidence Interval

We have also calculated the range of confidence interval, considering it 95% for each algorithm with 160 characters as input because the reported margin of error is typically about twice the standard deviation – the radius of a 95% confidence interval [31]. Confidence interval is an interval estimate of a population parameter and is used to indicate the reliability of an estimate. Fig. 6(a), 6(b), 6(c), 6(d) and 6(e) represent the range of confidence interval (high & low range values) for both encryption (E_low_interval, E_high_interval) and decryption (D_low_interval, D_high_interval) of the message (SMS) for 160, 320, 480, 640 and 800 characters in length for DES, Triple-DES with 2-keys, Triple-DES with 3-keys and

AES algorithms where all times are in nanoseconds. We have used t-distribution to calculate the confidence interval because it computes confidence intervals for large ‘n’ if the data is not normally distributed, i.e., for large ‘n’ the sample mean converges in distribution to a normal distribution and for large degrees of freedom the t-distribution converges to a normal distribution [32]. In this process, the SMS size from 160 to 800 characters is evaluated where more than 160 characters in an SMS is split and concatenated with another SMS. Thus, transmitted message can contain a range of 1120 to 56000 bits where each character is mapped with 7-bit ASCII value. A low standard deviation indicates that the data points tend to be very close to the mean, whereas high standard deviation indicates that the data points are spread out over a large range of values. Since, the AES algorithm is strict to its output range, thus, it is best among them.

C. A Variant of AES: MAES Algorithm

AES with 128-bit key has proved to be an efficient algorithm to encrypt the SMS but its security cannot be remain maintained in the subsequent years. Various researchers have found attacks on AES with 128-bit key [33], [34] with some assumptions. Thus, we propose a variant of AES called MAES (modified AES) which is more secure with 256-bit key (as original AES) and 256-bit each block of data. The increase in length of each block improves the performance of original AES. Various steps of the MAES algorithm are as follows:
(1) Key Generation: In EasySMS protocol, 256-bit of DK1 key is generated at the MS1 and AS which is used as cipher key for MAES and round keys are derived from this 256 bits cipher key using AES key schedule. (2) Initial Round: AddRoundKey—each byte of the state is combined with the round key using bitwise XOR. (3) Rounds: (i) SubBytes—a non-linear substitution step where each byte is replaced with another according to a lookup table, (ii) ShiftRows—a transposition step where each row of the state is shifted cyclically a certain number of steps, (iii) MixColumns—a mixing operation which operates on the columns of the state, (iv) AddRoundKey. (4) Final Round (no MixColumns): (i) SubBytes (ii) ShiftRows (iii) AddRoundKey

On considering the best assembly code combinations and continuance memory usage, the order of SubByte and ShiftRow processes are swapped, to reduce the number of times in memory reads and writes, as well as increase the computation speed without compromising the actual result [35], and this is done with MAES algorithm. Next, in AES, the MixColumns step is defined as a multiplication of columns with the matrix M. The matrix M used in the AES and its inverse matrix M^{-1}, both are different and the calculation of inverse of a matrix increases the computation. Thus, we used an alternative matrix M_1 because for new matrix, $M_1 = M_1^{-1}$.

$$M = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \quad M^{-1} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix}$$

$$M_1 = M_1^{-1} = \begin{bmatrix} 1 & 2 & 4 & 6 \\ 2 & 1 & 6 & 4 \\ 4 & 6 & 1 & 2 \\ 6 & 4 & 2 & 1 \end{bmatrix}$$

Table VI shows the performance of AES and MAES algorithms with one SMS size of plain text and cipher text pairs in bits and characters, where MAES generates 158 characters after ciphering the SMS of 160 characters. We have implemented various algorithms DES, Triple-DES, AES, CAST6, Twofish, RC2, RC6, MAES and performed the encryption/decryption of SMS with 160 characters which are shown in Fig. 5(c) and Fig. 5(d). Finally, we conclude that out of these algorithms, the MAES algorithm is more efficient to encrypt the SMS. The confidence interval for AES and MAES can be observed from Fig. 6(f) where confidence interval (high & low range values) of the MAES is strictly close to the encryption process.

VI. FORMAL PROOF OF PROPOSED PROTOCOL

In order to clear statement of our analysis, we use the BAN-Logic symbols to formally proof the authentication process of the proposed protocol. (1) $P \mid\equiv X$: P believes X, or P would be entitled to believe X, (2) $P \lhd X$: P sees X. Someone has sent a message containing X to P, who can read and repeat X, (3) $P \mid\sim X$: P once said X. P at some time sent a message including the statement X, (4) $P \mid\Rightarrow X$: P has jurisdiction over X. P is an authority on X and should be trusted on this matter, (5) $\#(X)$: The formula X is fresh, that is, X has not been sent in a message at any time before the current run of the protocol, (6) $P \overset{K}{\leftrightarrow} Q$: P and Q may use the shared key K to communicate, (7) $P \overset{X}{\Leftrightarrow} Q$: The formula X is a secret known only to P and Q, (8) $(X)_y$: This represents X combined with the formula Y that Y be a secret.

TABLE VI
SMS SIZE (INPUT, OUTPUT)

One SMS Size	AES	MAES
In Bits	1120, 1540	1120, 1111
In Characters	160, 220	160, 158

1) The formal messages in SAKA protocols: Phase-1: (1): $MS_1 \rightarrow MS_2 : ID_1, Ta, \mathrm{Re}\, qNo\ , \mathrm{f}_{1SK_1}(\mathrm{ID}_1 \,\|\, \mathrm{ReqNo})$; $MS_1 \overset{SK_1}{\leftrightarrow} AS_1$; (2): $MS_2 \rightarrow AS_2 : ID_1, ID_2, Tb, \mathrm{Re}\, qNo\ , \mathrm{f}_{1SK_1}(\mathrm{ID}_1 \,\|\, \mathrm{ReqNo})$, $\mathrm{f}_{1SK_2}(\mathrm{ID}_2 \,\|\, \mathrm{Tb} \,\|\, \mathrm{f}_{1SK_1}(\mathrm{ID}_1 \,\|\, \mathrm{ReqNo}))$; $MS_2 \overset{SK_2}{\leftrightarrow} AS_2$; (3): $AS_2 \rightarrow CA/RA : \{ID_1, ID_2, Tc\}_{SK_{AS-CA}}$; $\forall AS_i \overset{SK_{AS_i - CA}}{\leftrightarrow} CA$; (4): $CA/RA \rightarrow AS_2 : \{AS_1, Tc\}_{SK_{AS-CA}}$; (5): $AS_2 \rightarrow AS_1 : \{ID_1, \mathrm{Re}\, qNo, \mathrm{f}_{1SK_1}(\mathrm{ID}_1 \,\|\, \mathrm{ReqNo})\}_{SK_{AS_1 - AS_2}}$; $\forall AS_i \overset{SK_{AS_i - AS_j}}{\leftrightarrow} \forall AS_j, where i \neq j$; (6):

$AS_1 \rightarrow MS_1 : Td, Exptime, \mathrm{f}_{1\mathrm{SK}_1}(\mathrm{Td} \,\|\, \mathrm{Exptime} \,\|\, \mathrm{ReqNo})$; (7): $MS_1 \rightarrow AS_1 : \{Td, \mathrm{ReqNo}\}_{\mathrm{DK}_1}$; $MS_1 \overset{DK_1}{\leftrightarrow} AS_1$; (8): $AS_1 \rightarrow AS_2 : \{\mathrm{Re}\,qNo, Exptime, \mathrm{f}_{2\mathrm{SK}_1}(\mathrm{Td} \,\|\, \mathrm{ReqNo})\}_{\mathrm{SK}_{\mathrm{AS_1\text{-}AS_2}}}$ (9): $AS_2 \rightarrow MS_2 : \{Te, \mathrm{Re}\,qNo, Exptime, \mathrm{f}_{2\mathrm{SK}_1}\{(\mathrm{Td} \,\|\, \mathrm{ReqNo})\}_{\mathrm{SK}_{\mathrm{AS_1\text{-}AS_2}}}\}_{SK_2}$; (10): $MS_2 \rightarrow AS_2 : \{Te\}_{\mathrm{SK}_2}$; (11): $MS_2 \rightarrow MS_1 : \{Ta, \mathrm{Re}\,qNo\}_{\mathrm{DK}_1}$

Phase-2: (1): $MS_1 \rightarrow MS_2 : \{T_i, ID_1\}_{\mathrm{DK}_1}$; (2): $MS_2 \rightarrow MS_1 : \{T_i\}_{\mathrm{DK}_1}$

2) Security Assumptions: (a). It is assumed that SK is a secure key which is shared between MS and AS. (1) MS has SK key and $MS \mid\equiv MS \overset{SK}{\leftrightarrow} AS$, (2) AS has SK key and $AS \mid\equiv MS \overset{SK}{\leftrightarrow} AS$; (b). It is assumed that AS trusts the CA/RA through a secret key. $CA/RA \mid\equiv CA/RA \overset{SK_{CA-AS}}{\leftrightarrow} AS$ and $AS \mid\equiv CA/RA \overset{SK_{CA-AS}}{\leftrightarrow} AS$; (c). It is assumed that communication between all AS are done with a secret key shared between each pair of AS, i.e., $AS_i \mid\equiv AS_i \overset{SK_{AS1-AS2}}{\leftrightarrow} AS_j$ and $AS_j \mid\equiv AS_j \overset{SK_{AS1-AS2}}{\leftrightarrow} AS_i$,, where $i \neq j$.

3) Security Analysis: Phase-1: (1): $MS_1 \rightarrow MS_2 : MS_1 \mid\equiv \#(Ta) \wedge AS_1 \mid\equiv \#(Ta); MS_2 \triangleleft ID_1, Ta, \mathrm{Re}\,qNo$, $\mathrm{f}_{1\mathrm{SK}_1}(\mathrm{ID}_1 \,\|\, \mathrm{ReqNo})$; $MS_1 \overset{SK_1}{\leftrightarrow} AS_1$; (2): $MS_2 \rightarrow AS_2 : MS_2 \mid\equiv \#(Tb) \wedge AS_2 \mid\equiv \#(Tb); AS_2 \triangleleft ID_1, ID_2, Tb, \mathrm{Re}\,qNo$, $\mathrm{f}_{1\mathrm{SK}_1}(\mathrm{ID}_1 \,\|\, \mathrm{ReqNo}), \mathrm{f}_{1\mathrm{SK}_2}(\mathrm{ID}_2 \,\|\, \mathrm{Tb} \,\|\, \mathrm{f}_{1\mathrm{SK}_1}(\mathrm{ID}_1 \,\|\, \mathrm{ReqNo}))$; $MS_2 \overset{SK_2}{\leftrightarrow} AS_2$; (3): On receiving, the AS_2 calculates $\mathrm{f}_{1\mathrm{SK}_2}(\mathrm{ID}_2 \,\|\, \mathrm{Tb} \,\|\, \mathrm{f}_{1\mathrm{SK}_1}(\mathrm{ID}_1 \,\|\, \mathrm{ReqNo}))$, if it matches then $AS_2 \rightarrow CA/RA : \{ID_1, ID_2, Tc\}_{SK_{AS-CA}}$; $\forall AS_i \overset{SK_{AS_i-CA}}{\leftrightarrow} CA$, (4): After receiving the message from AS_2 the CA/RA validate ID_1 and ID_2 and then $CA/RA \rightarrow AS_2 : \{AS_1, Tc\}_{SK_{AS-CA}}$; (5): $AS_2 \rightarrow AS_1 : \{ID_1, \mathrm{Re}\,qNo, \mathrm{f}_{1\mathrm{SK}_1}(\mathrm{ID}_1 \,\|\, \mathrm{ReqNo})\}_{SK_{AS_1-AS_2}}$; $\forall AS_i \overset{SK_{AS_i-AS_j}}{\leftrightarrow} \forall AS_j, where\; i \neq j$; (6): First AS_1 computes $\mathrm{f}_{1\mathrm{SK}_1}(\mathrm{ID}_1 \,\|\, \mathrm{ReqNo})$ then $AS_1 \rightarrow MS_1 : Td, Exptime, \mathrm{f}_{1\mathrm{SK}_1}(\mathrm{Td} \,\|\, \mathrm{Exptime} \,\|\, \mathrm{ReqNo})$; (7): The MS_1 computes $\mathrm{f}_{1\mathrm{SK}_1}(\mathrm{Td} \,\|\, \mathrm{Exptime} \,\|\, \mathrm{ReqNo})$ and compares it with the received one, then $MS_1 \rightarrow AS_1 : \{Td, \mathrm{ReqNo}\}_{\mathrm{DK}_1}$; $MS_1 \overset{DK_1}{\leftrightarrow} AS_1$; (8): AS_1 checks ReqNo and #Td then $AS_1 \rightarrow AS_2 : \{\mathrm{Re}\,qNo, Exptime, \mathrm{f}_{2\mathrm{SK}_1}(\mathrm{Td} \,\|\, \mathrm{ReqNo})\}_{\mathrm{SK}_{\mathrm{AS_1\text{-}AS_2}}}$; (9): $AS_2 \rightarrow MS_2 : \{Te, \mathrm{Re}\,qNo, Exptime, \mathrm{f}_{2\mathrm{SK}_1}(\mathrm{Td} \,\|\, \mathrm{ReqNo})\}_{\mathrm{SK}_{\mathrm{AS_1\text{-}AS_2}}}\}_{SK_2}$ (10): $MS_2 \rightarrow AS_2 : \{Te\}_{\mathrm{SK}_2}$ and checks #Te with the received #Te; (11): $MS_2 \rightarrow MS_1 : \{Ta, \mathrm{Re}\,qNo\}_{\mathrm{DK}_1}$, if MS_1 finds correct #Ta and ReqNo then the authentication is successful.

Phase-2: (1): $MS_1 \rightarrow MS_2 : \{T_i, ID_1\}_{\mathrm{DK}_1}$; On receiving the message the MS_2 checks validity of ID_1 and $T_i \leq Exptime$. (2): $MS_2 \rightarrow MS_1 : \{T_i\}_{\mathrm{DK}_1}$; If received T_i is same as was sent then authentication is completed.

4) Message meaning Rule: (1) $\dfrac{MS_1 \mid\equiv (MS_1 \overset{DK_1}{\leftrightarrow} MS_2) \wedge (MS_2 \overset{SK_2}{\leftrightarrow} AS_2) \wedge (MS_1 \overset{SK_1}{\leftrightarrow} AS_1), AS_2 \triangleleft \mathrm{f}_{1\mathrm{SK}_2}(\mathrm{ID}_2 \,\|\, \mathrm{Tb} \,\|\, \mathrm{f}_{1\mathrm{SK}_1}(\mathrm{ID}_1 \,\|\, \mathrm{ReqNo}))}{MS_2 \mid\equiv AS_2 \mid\sim \mathrm{f}_{1\mathrm{SK}_2}(\mathrm{ID}_2 \,\|\, \mathrm{Tb} \,\|\, \mathrm{f}_{1\mathrm{SK}_1}(\mathrm{ID}_1 \,\|\, \mathrm{ReqNo}))}$

(2) $\dfrac{AS_1 \mid\equiv \mathrm{f}_{2\mathrm{SK}_1}(\mathrm{Td} \,\|\, \mathrm{ReqNo}) \wedge (AS_1 \overset{SK_1}{\leftrightarrow} MS_1), MS_1 \triangleleft \mathrm{f}_{1\mathrm{SK}_1}(\mathrm{Td} \,\|\, \mathrm{Exptime} \,\|\, \mathrm{ReqNo})}{AS_1 \mid\equiv MS_1 \mid\sim \mathrm{f}_{1\mathrm{SK}_1}(\mathrm{Td} \,\|\, \mathrm{Exptime} \,\|\, \mathrm{ReqNo})}$

5) Nonce/Timestamp Verification Rule: (1) $\dfrac{MS_1 \mid\equiv \#(Ta) \wedge MS_2 \mid\equiv \#(Tb), MS_2 \mid\equiv AS_2 \mid\sim \mathrm{f}_{1\mathrm{SK}_2}(\mathrm{ID}_2 \,\|\, \mathrm{Tb} \,\|\, \mathrm{f}_{1\mathrm{SK}_1}(\mathrm{ID}_1 \,\|\, \mathrm{ReqNo}))}{MS_2 \mid\equiv AS_2 \mid\equiv \mathrm{f}_{1\mathrm{SK}_2}(\mathrm{ID}_2 \,\|\, \mathrm{Tb} \,\|\, \mathrm{f}_{1\mathrm{SK}_1}(\mathrm{ID}_1 \,\|\, \mathrm{ReqNo}))}$

(2) $\dfrac{AS_2 \mid\equiv \#((Tc) \wedge \#(Te)) \wedge AS_1 \mid\equiv \#(Td), AS_1 \mid\equiv MS_1 \mid\sim \mathrm{f}_{1\mathrm{SK}_1}(\mathrm{Td} \,\|\, \mathrm{Exptime} \,\|\, \mathrm{ReqNo})}{AS_1 \mid\equiv MS_1 \mid\equiv \mathrm{f}_{1\mathrm{SK}_1}(\mathrm{Td} \,\|\, \mathrm{Exptime} \,\|\, \mathrm{ReqNo})}$

6) Jurisdiction Rule: (1) $\dfrac{MS_2 \mid\equiv AS_2 \Rightarrow \mathrm{f}_{1\mathrm{SK}_2}(\mathrm{ID}_2 \,\|\, \mathrm{Tb} \,\|\, \mathrm{f}_{1\mathrm{SK}_1}(\mathrm{ID}_1 \,\|\, \mathrm{ReqNo})), MS_2 \triangleleft AS_2 \mid\sim \mathrm{f}_{1\mathrm{SK}_2}(\mathrm{ID}_2 \,\|\, \mathrm{Tb} \,\|\, \mathrm{f}_{1\mathrm{SK}_1}(\mathrm{ID}_1 \,\|\, \mathrm{ReqNo}))}{MS_1 \mid\equiv MS_2 \mid\equiv AS_2 \mid\equiv AS_1}$

(2) $\dfrac{AS_1 \mid\equiv MS_1 \Rightarrow \mathrm{f}_{1\mathrm{SK}_1}(\mathrm{Td} \,\|\, \mathrm{Exptime} \,\|\, \mathrm{ReqNo}), AS_1 \triangleleft MS_1 \mid\sim \mathrm{f}_{1\mathrm{SK}_1}(\mathrm{Td} \,\|\, \mathrm{Exptime} \,\|\, \mathrm{ReqNo})}{(AS_1 \mid\equiv MS_1) \wedge (AS_2 \mid\equiv MS_2) \mid\equiv AS_2 \mid\equiv MS_1}$

7) Protocol Goals: (a) Mutual Authentication between the MS and the AS: $MS_2 \mid\equiv AS_2 \wedge AS_1 \mid\equiv MS_1 \rightarrow MS_1 \mid\equiv MS_2 \mid\equiv AS_2 \mid\equiv AS_1$, thus mutual authentication is hold.

(*b) Efficient Key Management between sender and receiver MS:* There is one key DK_1 between the MS and the AS to provide agreement. $AS_1 \mid\equiv \#(Td), MS_1 \mid\equiv DK_1 \wedge \#(Td)$, since $DK_1 = \mathrm{f}_{2\mathrm{SK}_1}(\mathrm{Td} \,\|\, \mathrm{ReqNo})$; $AS_2 \mid\equiv \#(Te), MS_2 \mid\equiv SK_2 \wedge \#(Te)$, and $(AS_1 \rightarrow AS_2) \wedge (AS_2 \rightarrow MS_2) \mid\sim DK_1$, *(c) Key Freshness between the MS and the AS:* $AS_1 \mid\equiv \#(Td) \wedge MS_1 \mid\equiv \#(Td), AS_2 \mid\equiv \#(Te) \wedge MS_2 \mid\equiv \#(Te)$, $DK_1 = \mathrm{f}_{2\mathrm{SK}_1}(\mathrm{Td} \,\|\, \mathrm{ReqNo})$, (*d*) *Confidentiality between the end-to-end MS via AS:*

$$\frac{MS_1 \mid\equiv (MS_1 \overset{DK_1}{\leftrightarrow} MS_2), MS_2 \triangleleft \{Msg\}_{DK_1}}{MS_1 \mid\equiv MS_2 \mid\sim Msg} \wedge$$

$$\frac{MS_2 \mid\equiv (MS_2 \overset{DK}{\leftrightarrow} MS_1), MS_1 \triangleleft \{Msg\}_{DK_1}}{MS_2 \mid\equiv MS_1 \mid\sim Msg}$$

(e) Resistance Replay Attack: If the attacker gets #Ta from message (1) and #Tb from message (2), he/she is unable to forge the message because he/she doesn't know SK_1 and SK_2. If the attacker gets #Td from message (6) and #Te from message (9), he/she is unable to forge the message because he/she doesn't know DK_1 and SK_2. Since #Ta, #Tb, #Td and #Te will be changed next time thus it resistance the attack. *(f) Resistance Man-in-the-Middle Attack:* Since attacker neither knows DK_1 nor $\{\}_{DK_1}$ encryption algorithm, thus it prevents the communication from being eavesdropped. (*g*) *Resistance SMS Disclosure and OTA Attack:* The MAES algorithm is proposed to use as $\{\}_{DK_1}$ which prevents SMS disclosure attack. End-to-end security of message OTA between both MS is provided by MAES with DK_1.

h. Resistance Impersonation Attack: (1) Adversary tries to impersonate MS: Since $\mathrm{f}_{1\mathrm{SK}_2}(\mathrm{ID}_2 \parallel \mathrm{Tb} \parallel \mathrm{f}_{1\mathrm{SK}_1}(\mathrm{ID}_1 \parallel \mathrm{ReqNo}))$ and $\mathrm{f}_{1\mathrm{SK}_1}(\mathrm{ID}_1 \parallel \mathrm{ReqNo})$ are computed at MS_2 and MS_1, and compared at AS_2 and AS_1 respectively. This prevents the MS from the impersonation attack. *(2) Adversary tries to impersonate AS:* The integrity value $\mathrm{f}_{1\mathrm{SK}_2}(\mathrm{ID}_2 \parallel \mathrm{Tb} \parallel \mathrm{f}_{1\mathrm{SK}_1}(\mathrm{ID}_1 \parallel \mathrm{ReqNo}))$ at MS_2 and the AS_2 will be violated. Additionally, if the MS_1 receives $\mathrm{f}_{1\mathrm{SK}_1}(\mathrm{Td} \parallel \mathrm{Exptime} \parallel \mathrm{ReqNo})$ at any time, then the connection will be terminated because MS_1 has not sent any request.

VII. CONCLUSION

EasySMS protocol is successfully designed in order to provide end-to-end secure communication through SMS between mobile users. The analysis of the proposed protocol shows that the protocol is able to prevent various attacks. The transmission of symmetric key to the mobile users is efficiently managed by the protocol. This protocol produces lesser communication and computation overheads, utilizes bandwidth efficiently, and reduces message exchanged during authentication than SMSSec and PK-SIM protocols.

ACKNOWLEDGMENT

This research work is supported by TCS, India.

REFERENCES

[1] Press Release. (2012, Dec. 3). Ericsson Celebrates 20 Years of SMS.

[2] R. E. Anderson et al., "Experiences with Transportation Information System that Uses Only GPS and SMS," IEEE ICTD, No. 4, 2010.

[3] D. Risi, M. Teófilo, "MobileDeck: Turning SMS into a Rich User Experience," 6th MobiSys, No. 33, 2009.

[4] Kuldeep Yadav, "SMSAssassin: Crowdsourcing Driven Mobile-based System for SMS Spam Filtering," Workshop Hotmobile, 2011, pp. 1-6.

[5] J. Chen, L. Subramanian, E. Brewer, "SMS-Based Web Search for Low-end Mobile Devices," 16th MobiCom, 2010, pp. 125-135.

[6] B. DeRenzi, "Improving Community Health Worker Performance through Automated SMS," 5th ICTD, 2012, pp. 25-34.

[7] M. Densmore, "Experiences with Bulk SMS for Health Financing in Uganda,"ACM CHI, 2012, pp. 383-398.

[8] J. Hellström, A. Karefelt, "Participation through Mobile Phones: A Study of SMS Use during the Ugandan General Elections 2011," ICTD, 2012, pp. 249-258.

[9] I. Murynets, R. Jover, "Crime Scene Investigation: SMS Spam Data Analysis," IMC, 2012, pp. 441-452.

[10] K. Par, "Smartphone remote lock and wipe system with integrity checking of SMS notification," IEEE ICCE, 2011, pp. 263-264.

[11] A. Nehra, R. Meena, "A robust approach to prevent software piracy," SCES, 2012, pp. 1-3.

[12] N. Gligoric, T. Dimcic, D. Drajic,"Application layer security mechanism for M2M communication over SMS," TELFOR, 2012, pp. 5-8.

[13] S. Gupta, S. Sengupta, M. Bhattacharyya, S. Chattrejee, B.S. Sharma, "Cellular phone based web authentication system using 3-D encryption technique under stochastic framework," AH-ICI, 2009, pp. 1-5.

[14] P. Traynor, W. Enck, P. McDaniel, T. Porta. (2009). Mitigating Attacks on Open Functionality in SMS-Capable Cellular Networks. *IEEE/ACM Tran. on Netw.*, 17(1), pp. 40-53.

[15] Y. Zeng, K. Shin, X. Hu, "Design of SMS Commanded-and-Controlled and P2P-Structured Mobile Botnets," WiSec, 2012, pp. 137-148.

[16] K. Hamandi, "Android SMS Botnet: A New Perspective," 10th ACM MobiWac, 2012, pp. 125-129.

[17] J. Lo, J. Bishop, J. Eloff (2008). SMSSec: An End-to-end Protocol for Secure SMS. *Computers & Security*, 27(5-6), pp. 154–167.

[18] H. Rongyu, Z. Guolei, C. Chaowen, X. Hui. (2009). A PK-SIM Card based End-to-end Security Framework for SMS. *Com. Standard & Interfaces*, 31, pp. 629-641.

[19] M. Hassinen, "Java based Public Key Infrastructure for SMS Messaging," ICTTA, 2006, pp. 88-93.

[20] S. Wu, C. Tan, "A High Security Framework for SMS," BMEI, 2009, pp. 1-6.

[21] A. Santis, A. Castiglione, U. Petrillo, "An Extensible Framework for Efficient Secure SMS," CISIS, 2010, pp. 843-850.

[22] M. Toorani, A. Shirazi, "SSMS-A secure SMS messaging protocol for the m-payment systems," IEEE ISCC, 2008, pp. 700–705.

[23] P. Mondal, P. Desai, S. Ghosh, "An Efficient SMS-Based Framework for Public Health Surveillance," IEEE PHT, 2013, pp. 244-247.

[24] C.C. Yang, Y.L. Tang, R.C. Wang. (2005). A secure & efficient authentication protocol for anonymous channel in wireless communications. *Applied Mathematics & Computation*, 169(2), pp. 1431–1439.

[25] Y. Khiabani, S. Wei, J. Yuan. (2012). Enhancement of Secrecy of Block Ciphered Systems by Deliberate Noise. *IEEE Tran. on IFS*, 7(5), pp. 1604-1613.

[26] S. Wei, J. Wang, R. Yin, J. Yuan. (2013, April). Trade-Off between Security and Performance in Block Ciphered Systems with Erroneous Ciphertexts. *IEEE Tran. on IFS*, 8(4), pp. 636-645.

[27] E. Biham, "Design Tradeoffs of the AES Candidates," Asiacrypt, LNCS, Springer-Verlag, 1998.

[28] Ray Rischpater. (2009). Beginning Java™ ME Platform. Messaging with Wireless API, part 4, pp. 373-407.

[29] E. Biham, A. Shamir. (1991). Differential Cryptanalysis of DES Cryptosystems. *J. of Crypto.*, 4(1), pp. 3–72.

[30] J. Choi1, J. Kim, J. Sung, S. Lee, J. Lim, "Related-Key and Meet-in-the-Middle Attacks on Triple-DES and DES-EXE," ICCSA, LNCS 3481, 2005, pp. 567–576.

[31] D. Altman, D. Machin, T. Bryant. (2000). Statistics with Confidence Intervals and Statistical Guidelines. BMJ Books, 254 pages.

[32] W. N. Venables, B.D. Ripley. (2002). Modern Applied Statistics with S. Springer, 4th Edition, XI, 497 pages.

[33] A. Biryukov, O. Dunkel., N. Keller.(2009). Key Recovery Attacks of Practical Complexity on AES Variants With Up To 10 Rounds. *Cryptol.*, pp. 1-17.

[34] Chong Hee Kim. (2012, Feb.). Improved Differential Fault Analysis on AES Key Schedule. *IEEE Tran. on IFS,* 7(1), pp. 41-50.

[35] C. F. Lu, Y. S. Kan, H. Chiang, C. Yang, "Fast Implementation of AES Cryptographic Algorithms in Smart Cards," IEEE 37th ICCST, 2003, pp. 573-579.

Neetesh Saxena (M'11) has done his undergraduation from Uttar Pradesh Technical University (UPTU) Lucknow, India and graduation from Guru Gobind Singh Indraprastha University (GGSIPU) Delhi, India. He is currently a PhD student in the Discipline of Computer science & Engineering at Indian Institute of Technology (IIT) Indore, India. His current research interests include Cryptography, Network Security, Mobile Computing and applications. He is a reviewer of various International Conferences and Journals including European Journal of Operation Research (EJOR), and International Journal of Network Security (IJNS). He is a member of several professional bodies including IEEE, ACM, and Computer Society of India (CSI).

Narendra S. Chaudhari (M'88-SM'10) has completed his undergraduate, graduate, and doctoral studies at Indian Institute of Technology (IIT), Mumbai, India, in 1981, 1983, and 1988 respectively.

Dr. Narendra has done significant research work on game AI, novel neural network models like binary neural nets and bidirectional nets, context free grammar parsing, and graph isomorphism problem. He has supervised more than 20 doctoral students and more than 80 Master's students. He has delivered invited talks and presented his research results in several countries like America, Australia, Canada, Germany, Hungary, Japan, United Kingdom, etc. He has delivered prestigious M.S. Ramanujam memorial lecture organized by The Institution of Engineers, India in the area of Computer Engineering. He has more than 250 publications in top quality international conferences and journals.

Dr. Narendra has shouldered many senior level administrative positions in Universities in India as well as abroad. A few notable assignments include: Dean - Faculty of Engineering Sciences, Devi Ahilya University, Indore, Member - Executive Council, Devi Ahilya University, Indore, Coordinator - International Exchange Program, Nanyang Technological University, Singapore, Deputy Director - GameLAB, Nanyang Technological University, Singapore. He was Dean - Research and Development, Indian Institute of Technology (IIT) Indore, and member, Board of Governors, IIT Indore since September 2010 to June 2013. Currently, he is Director of Visvesvarya National Institute of Technology, Nagpur, Maharashtra State, India.

He has been referee and reviewer for a number of premier conferences and journals including IEEE Transactions, Neurocomputing, etc. Dr. Narendra is *Fellow* and recipient of *Eminent Engineer* award (Computer Engineering) of The Institution of Engineers, India (IE- India), as well as *Fellow* of the Institution of Electronics and Telecommunication Engineers (IETE) (India), senior member of Computer Society of India, Senior Member of IEEE, USA, Member of Indian Mathematical Society (IMS), Member of Cryptology Research Society of India (CRSI), and many other professional societies.