International Journal of Research in Advanced Engineering and Technology
ISSN: 2455-0876
Impact Factor: RJIF 5.44
www.newengineeringjournal.in
Volume 4; Issue 2; May 2018; Page No. 52-57

A study and implementation of SMS security for business transactions

Shadab Haider, Dr. RK Singh
KNIT, Sultanpur, Uttar Pradesh, India

Abstract
This paper describes important related work towards the SMS Security and its implementation of Combined Algorithm (Digital Signature and SHA). The paper begins by describing the GSM security architecture which are applicable to almost every cellular network management. Then we describe the SIM Card security, RSA Digital Signature Algorithm, Secure Hash Algorithm. After describing the theoretical background we introduces the Problem statement and assumption and discusses the existing security keys and also discusses their flaws. At last in this paper we gives the proposed security keys and its security features and provides the role of Key Distribution Center (KDC) in the proposed work and also explains why RSA-1024 is used in the proposed work and also discusses how it can be effectively implemented using Binary method and Blakley's Method
We generate a Digital Signature for short messages using RSA-1024 to be used for secure transactions through mobile devices. These digitally signed documents can be used for various business transactions using GSM cellular network to preserve their authenticity, integrity and confidentiality. We assume that the entire current GSM cellular standard will provide a backbone to support the communication once digital signature has been generated.

Keywords: SMS Security, GSM security, algorithm, key distribution center (KDC)

1. Introduction

GSM-Security Architecture

Mobile wireless networks are more vulnerable to unauthorized access and eavesdropping when compared with the traditional fixed wired networks due to the mobility of users, wireless medium and the requirement of low power consumption by a mobile user. GSM provides a basic range of security features to ensure adequate protection for both the operator and customer. Currently, security in GSM consists of the following three aspects:

1. Authentication
2. Signal and Data Confidentiality
3. Anonymity

There are three proprietary algorithms used to achieve authentication and confidentiality A3, A5 and A8. A3 is used to authenticate the SIM for access to the network. A5 and A8 achieve confidentiality by encrypting the data to be sent over the networks. Anonymity means to protect the caller's identity and location which is achieved by the use of temporary identities (TMSI) instead of IMSI (International Mobile Subscriber Identity).

1.1 Authentication

Authentication is achieved using a basic challenge-response mechanism between the SIM and the network. A3 is implemented in the SIM card and the Authentication Center (AuC) or Home Location Register (HLR). A3 takes a 128 bit value Ki (subscriber i's specific authentication key) and 128 bit RAND random number (challenge sent by the network) as input data. It produces a 32 bit output value SRES, which is a Signed RESponse to the network challenge. The SIM and the network both have knowledge of Ki in such a way that Ki is not disclosed. The SIM must respond correctly to the challenge and to be authenticated and allowed to access the network. The authentication procedure is outlined in the following steps:

1. The MS send IMSI (or TMSI) to the network.
2. The network receives the IMSI and finds the corresponding Ki of that IMSI.
3. The network generates a 128 bit random number (RAND) and sent it to the MS over the air inter face.
4. The MS calculate a SRES with the A3 algorithm using the given challenge (RAND) and the key Ki residing in the SIM.
5. At the same time, the network calculates the SRES using the same algorithm and the same input.
6. The MS send the SRES to the network.
7. If SRES (network) equals SRES (MS), then SIM is authenticated and allowed to access the network.
8. If SRES (network) differs RES (MS), an authentication rejected signal is send to the SIM and access to the network is denied.

1.2 Confidentiality

Once the user has been successfully authenticated to the network, he can make calls and use the services he is subscribed to. It is necessary to encrypt the data that is to be transmitted over the network so that it cannot be used by the adversary. The algorithm used to encrypt the data to be transmitted, is called the ciphering algorithm A5. The key Kc used in this algorithm is generated by the cipher key generation algorithm A8. GSM makes use of a ciphering key to protect both data and signal on the vulnerable air interface.

After the user authentication, the RAND (delivered from the network) together with the Ki (from the SIM) is sent to the A8 ciphering algorithm, to produce a ciphering key Kc. The A8 algorithm is stored on the SIM card. The Kc created by the A8 algorithm, is then used with the A5 ciphering algorithm to encipher or decipher the data.

1.3 Anonymity

When a new GSM subscriber turns on its phone for the first time, its IMSI is transmitted to the AuC on the network. After which, a Temporary Mobile Subscriber Identity (TMSI) is assigned to the subscriber. To ensure subscriber identity confidentiality, the Temporary Mobile Subscriber Identity (TMSI) is used. The TMSI is sent to the mobile station after the authentication and encryption procedures have taken place. The mobile station responds by confirming reception of the TMSI.

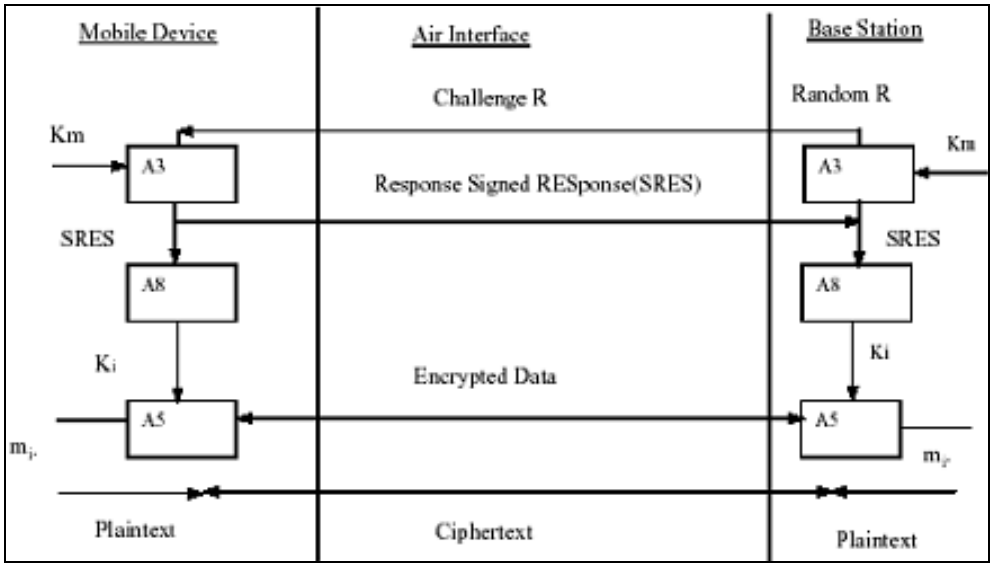

Fig 1: GSM Authentication and Ciphering

2. Security Implementation

Security parameters of the GSM mainly depend on Mobile Station (MS) and the GSM network. Mobile Station (MS) consists of two main elements: The Mobile Equipment (ME) and the Subscriber Identity Module SIM. The SIM card store the sensitive information such as the International Mobile Subscriber Identity (IMSI), Ki (a secret key for authentication), and other user information. All these information may be protected by personal identity number (PIN). This removable card can also store the information such as the users phone number, phone book as well as other information related to the subscriber. Within the GSM application the three primary roles of the SIM are, access control to the network (authentication and ciphering), service personalization (SMS, advice of charge, etc.), network branding and advertising (graphics printed on SIM card). The new generation SIMs will enable services such as virtual cash, mobile banking, ticket reservations etc.

As soon as an MS is connected, subscriber identity is checked using a secret key Ki which is kept on the SIM card and on the HLR. The owner identifies herself to the mobile phone, the hardware, by a personal identification number (PIN). The hard-ware has a unique International Mobile Equipment Identity number (IMEI) which could be used to identify and suspend stolen devices from service. To identify a subscriber to the service worldwide, the SIM card holds a code number called International Mobile Subscriber Identity (IMSI). The IMSI consists of 15 positions coded by 4 bits. It includes the country ID, the mobile network ID and the subscriber ID. The IMSI is radioed to the HLR via the next BST on the very first connection to the network only. The MS authenticates itself after receipt of the RAND value generating SRES by help of its key Ki accompanying the IMSI.

3. RSA Digital Signature Algorithms

RSA [3, 10, 16] is a public key cryptographic algorithm given by Rivest, Shamir, and Adelman and is used for encryption and digital signatures. RSA was developed in 1977 and is today the most commonly used encryption and authentication algorithm. The RSA algorithm provides a procedure for signing a digital document, and verifying whether the signature is indeed authentic. RSA Digital algorithm works same as RSA Encryption algorithm; only the difference is in applying the keys. For signing the document sender his private key which can be verified by public key. The signature is computed using rules and parameters such that the identity of the signer and the integrity of the signed data can be verified. To encrypt (sign) a message M with RSA, using a private encryption key (e; n), we proceed as follows. First, represent the message as an integer between 0 and n -1. (Break a long message into a series of blocks, and represent each block as such an integer). Use any standard representation. The purpose here is not to encrypt the message but only to get it into the numeric form necessary for encryption.

Then, encrypt the message by raising it to the eth power modulo n. That is, the result (the ciphertext C) is the remainder when me is divided by n. To decrypt the ciphertext, raise it to another power d, again modulo n.

RSA Algorithm

i) Key Generation Steps

a) Select p, q where p, q are large prime, p=q.
b) Compute n = p*q.
c) Compute f(n) = (p-1)*(q-1).
d) Find e such that gcd (e, f(n)) = 1; e lies between 1 and f(n).
e) Compute d such that e*d (mod n) = 1. Where e, n is the private key pair to generate digital signature (by encryption) and d, n is the public key pair by which signature is verified (by decryption).

ii) Signing the Message

C = (M) e (modn) (1)

iii) Verifying the message

M = (C) d (modn) (2)

4. SHA- A Secure Has Algorithm

Secure Hash Algorithm is iterative, one-way hash functions that can process a message to produce a condensed representation called a message digest. SHA-256 produces a message of 256-bit for the message of size less than 264 bits. This algorithm enables the determination of a message's integrity: any change to the message will, with a very high probability, produce different message digest. This property is useful in the generation and verification of digital signatures and message authentication codes, and in the generation of random numbers (bits).

The algorithm can be described into two stages: preprocessing and hash computation. Preprocessing involves padding a message, parsing the padded message into m-bit blocks, and setting initialization values to be used in the hash computation.

The hash computation generates a message schedule from the padded message and uses that schedule, along with functions, constants, and word operations to iteratively generate a series of hash values. The final hash value generated by the hash computation is used to determine the message digest.

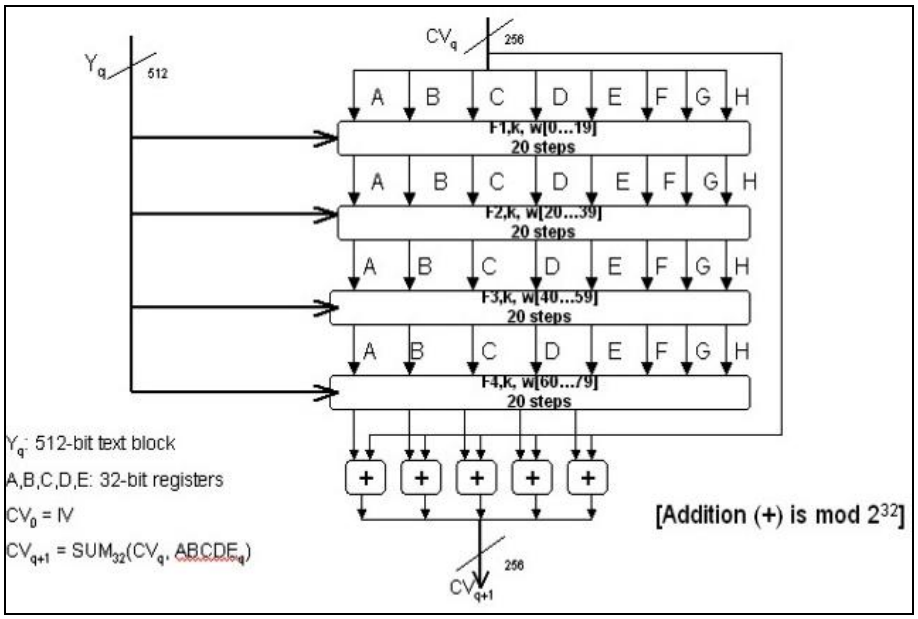

Fig 2: SHA-256 Computation

5. Existing Security keys

At present, a secret key Ki (128 bit) is stored in the SIM and is used for subscriber identity. This key (Ki) is used along with other algorithms and values to ensure the security as discussed below. The process is initiated by the user wanting to make a call from his mobile. The Visitor Location Register (VLR) establishes the identity of the SIM. This is determined through a 5 digit temporary identity number known as the Temporary Mobile Subscriber Identity (TMSI). The TMSI is used in place of the International Mobile Subscriber Identity (IMSI) for security reasons. The VLR sends a request for authentication to the Home Location Register (HLR). This request will contain the SIMs IMSI (as the IMSI and the related TMSI should be stored in the VLR). The HLR generates a 128 bit random RAND challenge and sends it to the MS via the VLR. Using Ki (128 bits) which is stored in the HLR and RAND (128 bits), the HLR then calculates (SRES) (HLR) (32 bits) using the A3 authentication algorithm. SRES (HLR) is then sent to the VLR. Using Ki (128 bits) which is stored in the SIM and RAND (128 bits) that is received as a challenge, the SIM calculates SRES(SIM) (32 bits) using the A3 authentication algorithm. SRES (SIM) is then sent to the VLR. If SRES (HLR) = SRES (SIM), then the SIM is authenticated and allowed access to the network. If SRES (HLR) is not equal to SRES (SIM), an authentication rejected signal is sent to the SIM and access to the network is denied.

A number of weaknesses exist with existing security key (Ki). The main problem with GSM lies in a particular implementation of the A3/A8 authentication and cipher key generation algorithm COMP128 as discussed below [4; 14]. The mobile phone is paged by its TMSI to establish a radio connection. Once the connection is established, the attacker sends a request for the IMSI. The attacker can then keep

challenging the MS with carefully chosen RANDs so as to exploit flaws in the COMP128 algorithm. To each RAND the mobile phone will respond with a different SRES, which the attacker will collect and store. This process will be repeated until the attacker has gained enough information to learn Ki. Now the attacker has Ki and IMSI in their possession. This enables an attacker to impersonate the user, and make and receive calls and SMS messages in their name. They can also eavesdrop, since RANDs from the legitimate network to the legitimate user can be monitored, and thus combined with the known Ki can be used to determine the Kc used for voice and signaling data encryption. These shortcomings force us to use another key by which a more secure communication could be conducted using GSM cellular network.

6. Proposed Security Keys

Ks: Proposed key for signing message - Private Key

Kv: Proposed key for verifying message -Public Key

Every SIM is provided with Ks, by which subscriber's messages will be signed. These signed documents consist of original messages and corresponding digital signature. These signed documents can be sent over GSM network to conduct transactions ensuring security. At the receiver end, receiver will send a request to get corresponding Kv, to Global Database of keys Kv. Now applying the Kv on original message, the receiver will compare the signed message and original message. The KDC is responsible to maintain a database of public keys by which signature can be verified.

7. Key Distribution Center (KDC)

A Key Distribution Center enables secure communications among the groups of users in a network by providing common keys that can be used with a symmetric encryption algorithm to encrypt and decrypt messages the users wish to send to each other. A KDC is a network service that provides tickets and temporary session keys. The KDC maintains a database of principal names (users and services) and their associated secret keys. It is composed of the authentication server and the ticket granting ticket server. In cryptography, a key distribution center (KDC) is part of a system intended to reduce the risks inherent in exchanging keys. A typical operation with a KDC involves a request from a user to use some service. The KDC will use cryptographic techniques to authenticate requesting users as themselves. It will also check whether an individual user has the right to access to the service requested. If the authenticated user meets all prescribed conditions, the KDC can issue a ticket permitting access. KDC mostly operate with symmetric encryption, but to make it more secure we can use asymmetric encryption as we proposed. We should always use a secure machine to act as our KDC because if an unauthorized user gained access to the KDC, our security will penetrated. In the proposed work, KDC is responsible for generating pair of secret keys (Ks, Kv). Ks is stored in subscriber identity module along with subscriber's information. Kv is stored and maintained in a global database. Kv is produced on demand for signature verification.

8. Hashed RSA-1024 Digital Signature Algorithm

The security of many cryptographic schemes and protocols depends on the hardness of finding the factors of large integers drawn from an appropriate distribution. To determine what key sizes are appropriate for a given application, one needs concrete estimates for the cost of factoring integers of various sizes. Predicting these costs is difficult, for two reasons. First, the performance of modern factoring algorithms is not understood very well. Second, even when the exact algorithmic complexity is known, it is hard to estimate the concrete cost of a large-scale computational effort using current technology.

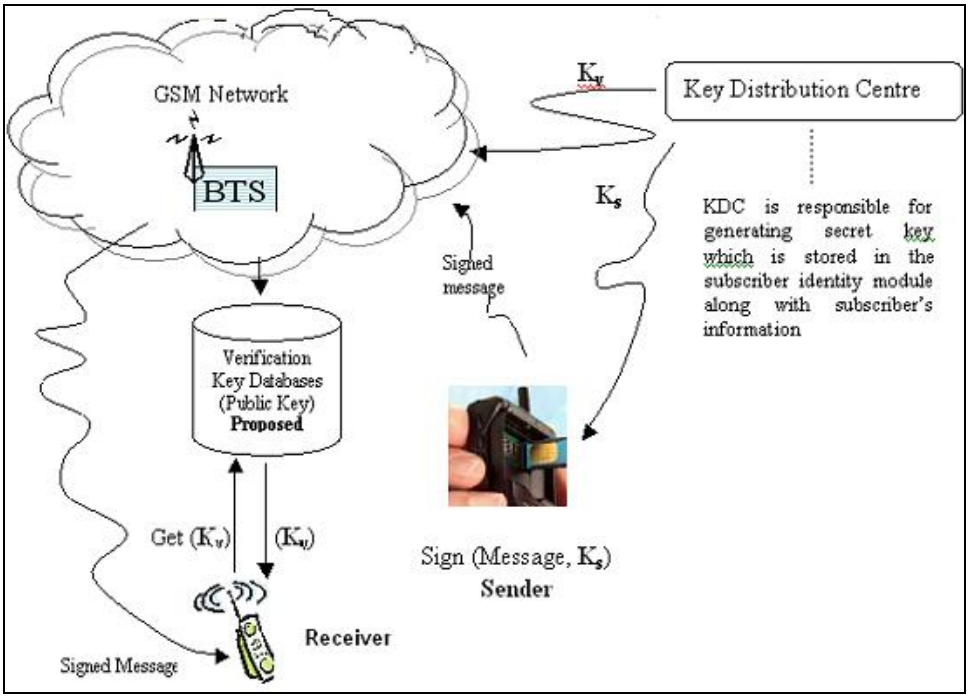

Fig 3: The Proposed KDC

Due to these difficulties, common practice is to rely on extrapolations from past factorization experiments. Many such experiments have been performed and published; for example, the successful factorization of a 512-bit RSA key in 1999 [3, 16] clearly indicated the insecurity of such keys for many applications, and forced us to use 1024-bit keys. It has been often claimed that 1024-bit RSA keys are safe for the next 15 to 20 years. In mathematics, RSA-1024 [3, 16] is one of the RSA numbers, large primes that are part of the RSA factoring Challenge. RSA-1024 has a length of 309 decimal digits and has not been factored so far. One of the most important challenges before RSA-1024 is that, encryption is quite slow because of large key size and modular exponentiation operation that have to use to ensure security. For the same reason, RSA's digital signature is slow as well. Another challenge is that length of transmitted signature equals the length of transmitted message. In this scheme, instead of signing a document, the hash of document is taken. The proposed scheme uses Secure Hash Algorithm (SHA-256) to obtain condensed version of message, which will go as input RSA Digital signature algorithm. Message is signed with sender's private key that can be verified by the receiver using sender's public key. The public exponent in the RSA algorithm is usually much smaller than the private exponent. In this way verification of a signature becomes faster than signing. This is desirable because a message will be signed by an individual only once, but the signature may be verified many times. To make it faster modified Hashed-RSA Algorithm is presented as following.

1. Obtain the message from the source (M).
2. Apply SHA-256 on M to get message digest (HM) as: HM = SHA-256(M).
3. Apply RSA-1024 Algorithm on this message digest to get digital signature (D) as: D = RSA-1024(HM); here we use private key for message signing.

Now these signed massage i.e. M+D can be send over GSM network using SM-SC to a particular user. At receiving end we can verify the integrity and authenticity of message /signature as follows:

1. Receive the signed message (M+D) from the sender.
2. Now again apply RSA Algorithm on Digital Signature (D). HM = RSA (D); here we use public key for message verification.
3. Again apply SHA-256 on the received massage to get HM' as: HM' = SHA-256(M).

Now message authenticity can be easily verified by comparing the values of HM and HM' as shown in figure 4.

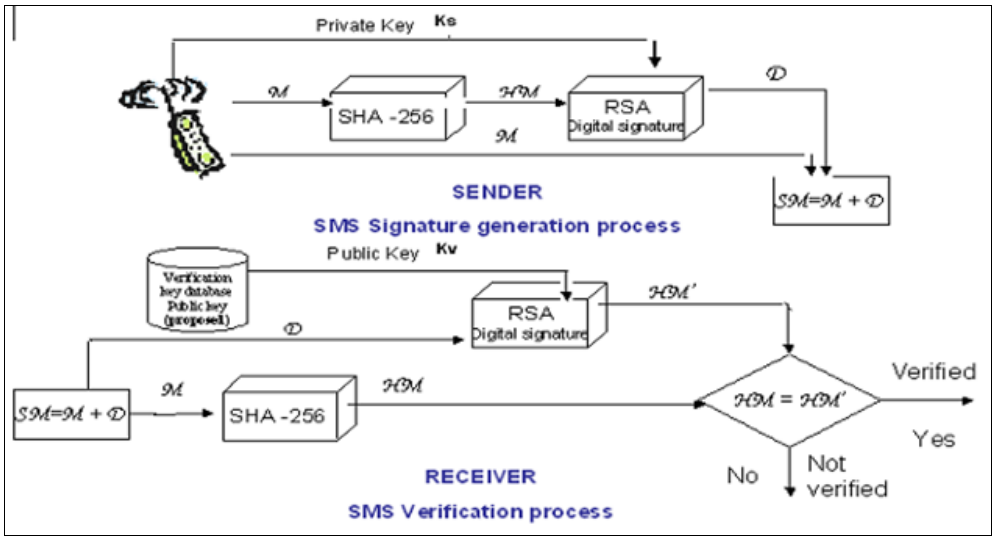

Fig 4: The Proposed SMS Signature Scheme

8.1 L-R/R-L Binary method

The L-R/R-L Binary method [9, 10] are used for modular exponentiation where we do not compute C: = me (mod n) by first exponentiating me and then performing a division to obtain the remainder C: = (Me) mod n. We have to perform numerous modular multiplications to compute me mod n. A different way of computing C = me (mod n) is to start with C: = M (mod n) and keep performing the modular multiplication operations C: = C *M (mod n) until C = me (mod n) is obtained. This approach requires e-1 modular multiplications to compute C: = me (mod n). This is known as binary method for modular exponentiation.

The binary method scans the bits of the exponent either from left to right or from right to left. A squaring is performed at each step, and depending on the scanned bit value, a subsequent multiplication is performed. In this way two variation of binary method are: L-R Binary and R-L Binary. We describe the left-to-right binary method for computing me (mod n) given the integers M, e, and n below. The right-to-left algorithm requires one extra variable to keep the powers of M.

LR Binary Method

Input: M; e; n
Output: C: = Me mod n

1. if eh-1 = 1 then C: = M else C: = 1
2. for i = h-2 downto 0
   2a. C: = C * C (mod n)
   2b. if ei = 1 then C: = C *M (mod n)
3. return C

If e = 250 = (11111010), which implies h = 8. Initially, we take C: = M since eh-1=e7=1.The number of modular multiplications required by the binary method for computing (M) (250) is found to be 7+5 = 12. For an arbitrary h-bit number e with ek-1=1, the Binary method requires:

1. Squaring (Step 2a): h-1 where h is the no. of bits in the binary expansion of e.
2. Multiplication (Step 2b): H (e)-1 where H (e) is the Hamming Weight (the no of 1s in the binary expansion) of e.

8.2 Blakley's Method

Blakley's method [9, 10] is used for modular multiplication which directly computes a* b mod n by interleaving the shift-add steps of the multiplication and the shift subtract steps of the division. Since the division algorithm proceeds bit-by-bit, the steps of the multiplication algorithm must also follow this process. This implies that we use a bit-by-bit multiplication algorithm rather than a word-by-word multiplication algorithm, which would be much quicker. However, the bit-by-bit multiplication algorithms can be made run faster by employing bit-recoding techniques. Furthermore, the m-ary segmentation of the operands and canonical recoding of the multiplier allows much faster implementations. In the following we describe the steps of Blakley's algorithm. Blakley's algorithm is based on the above formulation of the product t, however, at each step, we perform a reduction in order to make sure that the remainder is less than n. The reduction step may involve several subtractions.

Blakley's Algorithm

Input: a; b; n

1. Output: R = a.b mod n 1. R: = 0
2. For i = 0 to k-1
3. R: = 2R + ak -1- i^b
4. R: = R mod n
5. Return R

At Step 3, the partial remainder is shifted one bit to the right and the product ak-1-i^b is added to the result. This is a step of the right-to-left multiplication algorithm.

9. References

1. Midp apis for wireless applications. http://java.sun.com/ products/sjwtoolkit.
2. Secure hash signature standard (shs) (¯ps pub 180-2). http://csrc.nist.gov/cryptval/.
3. Eran Tromer Adi Shamir. On the cost of factoring rsa-1024. pages 187{193,2004. shamir,tromer@wisdom.wei zmann.ac.il.
4. Levent Ertaul Basar Kasim.
5. Cooke JC, Brewster RL. Cryptographic security techniques for mobile telephones. IEEE, 1992, 425-428.
6. Tal Rabin Jee Hea an, Yevgeniy Dodis. On the security of joint signature and encryption. Springer-Verlag, 2002, 83-107.
7. Palo Alto Jon Callas, PGP Corporation. Identity-based encryption with conventional public-key infrastructure, 2005.
8. Vorapranee Khu-smith and Chris J. Mitchell. Enhancing e-commerce security using gsm authentication. Technical Report OSU-CISRC-11 04-TR60, 2002.
9. Cetin Kaya Koc. High-speed rsa implementation. Technical report, 100 Marine Parkway, Suite 500, Redwood City, CA 94065-1031.
10. Cetin Kaya Koc. Rsa hardware implementation, 1995.
11. Chang-Tien Lu Lily R. Liang, Seema Nambiar. Analysis of payment transaction security in mobile commerce. Snambiar,ctlu@vt.edu,lliang@udc.edu.
12. Forum Nokia Midp. A brief introduction to secure sms messaging in midp. MIDP, 2003, 1-0.
13. Varsha Apte Nilesh Agarwal, Leena Chandran-Wadia. Capicity analysys of gsm short message service.
14. Laurent Gauteron Pierre Girard Helena Handschuh David Naccache Stephane Socie Claire Whelan Olivier Benoit, Nora Dabbous. Mobile terminal security. 1999. cwhelan@computing.dcu.ie.
15. Lauri Pesonen. Gsm interception. Lauri.Pesonen@iki.¯.
16. Shamir A, Rivest RL, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. ACM, 1978, 120-126.
17. Ronny Arild Tage Stabell-Kul, Per Harald Myrvang. Providing authentication to messages signed with a smart card in hostile environments, 1999. tage/ronnya/perm@ pasta.cs.uit.no.
18. Finn Trosby. Sms, the strange duckling of gsm, 2004, 187-193.
19. Wei Zhang. Gsm security issues. MIDP, 2000.