CRYPTOGRAPHY- AN IDEAL SOLUTION TO PRIVACY, DATA INTEGRITY AND NON-REPUDIATION

Getahun Mekuria, Eneyew Adugna, Deva Rajan
Electrical Engineering Department
Addis Ababa University

ABSTRACT

Information Secrecy deals with maintaining privacy through encryption, data integrity through hashing and non-repudiation through digital signatures; and has become the issue of today's communication engineering due to increasing failures to the secrecy of messages conveyed on a given networking system thereby creating damages beyond expectations.
Encryption, hashing and digital signatures are the three primitives of Cryptography and these have been treated in depth and their performances on text data and image data have been studied. The most secure algorithms so far in use have been introduced and the respective performance of each primitive's algorithm on text data and image dota have been studied. Performance differences for the two dota types have also been noted and outlined in detail.

INTRODUCTION

Establishing a fast and secure channel and facility is the quest of today's communication engineering for reliable and lasting service to the customer. The introduction and advancement of Networking system, be it LAN (Local Area Networking) or WAN (Wide area Networking), wired or wireless, and the progress of the Internet towards being a necessity for every one around the globe has increased the need for secure information interchange to minimize the problems caused by intruders. The basic motive behind the toiling of these intruders, known as *cryptoanalysts* [9], are industrial espionage, financial gains, revenge, terrorism and simple publicity.

People in the communication and computer engineering have really working to their best to introduce new security mechanisms on the one hand and to make the already existing security mechanisms more tightening both in software and hardware solutions, on the other hand. The intensity of these efforts have brought such secure encryption algorithm as Rivest Shamir and Adleman (RSA) encryption and digital signature algorithms for privacy and non-repudiation issues, respectively; Secure Hash Algorithm (SHA-1) for data integrity issue.

In Section-II the idea of maintaining privacy using RSA-Encryption algorithm has been treated and its performance on text data and image data has been studied. The issue of data integrity using SHA-1 is the subject of Section III, with the two data types in mind. In Section IV the idea of digital signature is introduced and RSA-Digital Signature Scheme is treated with results from performance differences on texts and images.

ENCRYPTION

Encryption is basically to maintain privacy and authentication of information interchange. Particularly the sender is assured that his message is read by non-other than whom he authorizes to read. A *private system* is a system in which an entity (a person, a computer terminal, pager, etc) is allowed to get a service to which it is authorized. The type of authorization may vary depending on the available potential or the limit of secrecy. Some entity may be authorized to read, some to modify, some to even create information. So, encryption is used for creating such a private system.

Encryption is a transformation whereby a message is turned to some unintelligent form using a secret value known as a *Key*. The sender of an information therefore encrypts his message and sends the unintelligent form of his message on a public channel to an intended recipient. The recipient performs the reverse transformation of encryption known as

Decryption using a key again. If the decryption key is the same as the encryption key used, or computationally simple given the encryption key, then such type of secrecy system is known as *Secret Key Cryptography*. In Secret Key Cryptography there must be a secret channel for the exchange of the key between the sender and the receiver, and this is the main problem of such type of secrecy system. However, in 1976 W. Diffie and M.E. Hellman from Sanford University [2] introduced a system in which the keys for encryption and decryption need not be equal and that knowledge of one of the two never leads to the computation of the other. This is taken care by a proper design of a *one-way function* [9] for the production of the key pairs. In this later secrecy system the encryption key is to be publicized and kept in a trusted communication directory and the decryption key is kept privately secret as long as the encryption key is in use. Any one who wants to send a message fetches the encryption key of the recipient from the trusted directory, encrypts a message with the key, and sends to the recipient. A message encrypted by a given encryption key never gets decrypted except with the corresponding decryption key. This type of secrecy system is known as *Public Key Cryptography*.

Since the introduction of Public Key Cryptography many researches have been made to find the appropriate one-way function and to establish a firm algorithm for production of the key pairs, and for encryption and decryption transformation algorithms in such a way that publicizing the encryption key never compromises the security of the system. The earliest and successful was the work of the three MIT Professors: Ronald Rivest, Adi Shamir and Leonard Adleman- which they named their system *RSA-cryptosystem* [7].

RSA was not only the earliest public key oriented cryptosystem, but also known to withstand so many years' cryptoanalytic attacks to date, of course with minor adjustments, and has since then become the standard of many secure information import and export policies around the world.

RSA CRYPTOSYSTEM

The mathematical foundation of RSA encryption/ decryption algorithms lies in the factorization difficulty of a composite number, which is a product of two very large prime numbers. For this reason two very large prime numbers must be generated with different prime number searching and generation techniques like the ones introduced in [7], [9].

Assume subscriber P_1 wants to be a member of some public key secure information communication group using the public network to which he has been connected. So, P_1 must first do the following to generate the key for encryption, *the public key*, and for decryption, *the private key*.

- Generates two prime numbers p and q of reasonably large and are roughly the same bit length, and that $p \quad q$ should not be too small.
- Computes the modulus $n = p.q$ and the Euler's totient function ϕ.

  Where
  $$\phi = (p-1) \cdot (q-1) \qquad (1)$$
- Selects a random integer e such that

  $$gcd(e, \phi) = 1 \qquad (2)$$
- Computes the unique integer d in the range

  $[0, \phi-1]$ such that
  $$e.d = 1 (\mathrm{mod}\ \phi) \qquad (3)$$
- Then subscriber P_1's public key is (n,e) and his private key is d. So, he publicizes (n,e) to a communication directory he best trusts.

Each subscriber of the communication system generates his own private and public key pairs and sends the public-key pair (n, e) to a trusted directory in the system so that any one looking for a reliable public-key pair of some other subscriber may be allowed access to the directory.

Assume a person P_2 wants to send a secure and private message to the subscriber P_1. At the same time P_2 wants to assure himself that his message, even if there are possibilities that it may fall in a wrong hand, is read by no one except the intended recipient, P_1.

So, P_2 does the following:

- Obtain P_1's public key, say (n_1, e_1), from the communication networks directory he trusts.
- Represent his message M in to smaller blocks m_1, m_2, ... Each of equal size of length l and lie in the range $[0, n\text{-}1]$.

$$M = m_1 \| m_2 \| m_3 \| \ldots\ldots \tag{4}$$

- For each small block m_i compute :

$$\left.\begin{aligned} c_1 &= m_1^{e1} \bmod n_1 \\ c_2 &= m_2^{e1} \bmod n_1 \\ c_3 &= m_3^{e1} \bmod n_1 \\ &\ldots\ldots \end{aligned}\right\} \tag{5}$$

- Then form a cipher message C from smaller blocks c_i computed for each message block m_i.

$$C = c_1 \| c_2 \| c_3 \| \ldots\ldots \tag{6}$$

- Send the cipher message C on a public channel in which P_1 is a subscriber.

Up on receiving the cipher message C from the communication channel P_1 does the following to decrypt C and get the original message M as follows using his private key, say d_1.

- Represents the message C in to smaller blocks c_1, c_2, ... each of equal size of length l and lie in the range $[0, n\text{-}1]$.
- For each small block c_i computes :

$$\left.\begin{aligned} m_1 &= c_1^{d1} \bmod n_1 \\ m_2 &= c_2^{d1} \bmod n_1 \\ m_3 &= c_3^{d1} \bmod n_1 \end{aligned}\right\} \tag{7}$$

- Then form the original message M from smaller blocks m_i computed for each message blocks c_i.

PERFORMANCE OF RSA ENCRYPTION FOR TEXT AND IMAGE DATA

Texts of considerably large sizes written in MS-DOS editor and images of the same sizes are encrypted and decrypted using RSA encryption and decryption algorithms [9]. The speed for the encryption and decryption of texts is found to be higher than the speed for encrypting and decrypting the images. That means the performance of RSA for text data is found to be faster than that of image objects. The reason for this result is to be discussed next.

Reason for slower performance of RSA on images

To implement the RSA encryption/decryption transformations each character is to be used to form integers represented by the word length of the computer used. In our case we used a computer which has a word length of 32 bits representing an integer of maximum size of $2^{32} - 1$ with big-endian architecture to change streams of characters to integers of 32-bits. In big-endian architecture, assume that b_1, b_2, b_3 and b_4 are four characters and we want to form a word W as:

$$W = b_1.2^{24} \mid b_2.2^{16} \mid b_3.2^{8} \mid b_4 \tag{8}$$

is the word formed with such an architecture.

So, four characters are to be combined in big-endian to form one 32-bit integer. Now, for the case of texts each character is to be represented by a value between 0 and 127 (i.e. 7-bits). Moreover the ASCII code of the numbers 0 through 9, the characters *a* through *z* and those *A* through *Z* all have so many 0 bits in their representation, and a 32-bit integer formed with such characters with big-endian architecture also has so many 0-bits. When the 32-bit numbers with many 0 bit is to be multiplied exponentially to bring the encryption and decryption transformations, it surely minimize the compntation thereby increasing the speed somehow, this is the case of text data encryption/decryption. But that was not the case for image data. An image element, pixel, can have any value between 0 and 255 (8-bits), and not confined to 0 through 127 numbers only like texts. This means that the probability of having characters with many 0 bits in image data is very much lesser. Forming 32-bit integers with the same big-endian architecture of such characters surely is not expected of producing integers of many 0 bits. Therefore, the exponential operations made at bits level, involving such integers is found to be very much slower in performance. This is the case

observed in the encryption and decryption transformation of the RSA cryptosystem for texts and image data.

A text message taken from the abstract of this paper, its encrypted and decrypted forms have been shown in Appendix-A. The encrypted form of the image in Appendix-B1 has been shown in B2, and the its decrypted form in B3.

DATA INTEGRITY THROUGH HASHING

The other issue of cryptography is to maintain the integrity of an information. That is the issue of checking any change or modification made in a data being conveyed on a certain public channel, which may include deletion, addition and change thereby altering the originally intended idea, and even may probably pass a message exactly opposite to what the sender sent. So there must be a system in which the receiver of an information is assured that the information is never changed on the channel, and be informed if there are any manipulations made. Dealing with such a case is what is commonly known as dealing with the problem of *data integrity*.

Data integrity and encryption are basically distinct. One may not mean the other, or one may not imply the other in the very strictest sense. That is why both of them are treated independently in the study of information security. As encryption is attained by functions and procedures of its own, data integrity is also attained by its own functions and procedures, and these functions that are used to attain the purpose and goal of data integrity are known as *Hashing Functions*, or simply *Hash Functions*.

Hash Functions take an input of a data of arbitrary bit length from an input space X and produce an output of fixed bit length known as *hash code*, *hash result* or simply *hash* by different cryptographers. For that reason a hash function properly designed for data integrity must have the following basic features.

Compression

A function h mapping an input of data X of length x_l with an output Y of length y_l must be able to produce a compression factor c given by

$$c = y_l/x_l \qquad (9)$$

such that $0 < c < 1$.

Ease of Computation

Given the hash function h and x, an element of the input space X, then it must be computationally easy to find y an element of the output space Y such that

$$y = h(x) \qquad (10)$$

Compression and ease of computation are the criteria a candidate function h is to be subject to before being selected as a hash function, there are also some basic properties the candidate function must fulfill before chosen as a hash function and hence used for data integrity.

Pre-image Resistant

For a pre-specified output element y in the output space Y, it must be computationally infeasible to find an element x in the input space X which maps to that particular y. This means that computation of

$$h(x) = y \qquad (11)$$

must be difficult.

2nd pre-image resistant

Given an input x which is an element of the input space X and y an element of the output space Y such that $y = h(x)$, then it must be computationally difficult to get a 2nd input x_2 in the message space X different from x such that

$$h(x_2) = y \qquad (12)$$

Producing the same out put as x.

Collision Resistant

It must be difficult to get two freely chosen inputs x and x_1 in the message space X which map to the same output. That is finding x and x_1 such that

$$h(x) = h(x_1) \qquad (13)$$

is difficult.

Here none of x or x_1 is pre specified as is the case in pre-image resistance.

HASHING

When someone says integrity of a given data is to be maintained it means that any manipulation made through out the data must be reflected at the final output by changing a code generated by the sender with the inherent compression facility available. For this to happen there must be iterative process which chains any two consecutive blocks of the message.

In Hashing the basic sections are the pre-processing and processing sections and for discussing them an image data of m by n pixel size is considered.

Preprocessing

For the discussion being made image data has been considered.

Given an image I of size m rows and n columns of pixels each represented with 8 bit gray level values. Then the size of this image is :

$$\text{Size of } (I) = m.n \text{ bytes} = 8.m.n \text{ bits} \quad (14)$$

Then in the preprocessing procedure this image I is represented by smaller blocks of each of which are r bits long and represented as $i_0, i_1, i_2, \ldots i_t$

$$\text{i.e.} \quad I = 8m.n \text{ bits} = r.t \text{ bits} \quad (15)$$

and therefore there must be t small blocks in the image I for some integer values of r and t.

where

$$t = 8m.n / r \quad (16)$$

But this is not always the case since $8m.n/r$ may not always produce an integer value. That is $8m.n$ may not always be an integer multiple of r. In such a case we have to perform another procedure under the preprocessing section known as *padding*. Padding is to append the image I with strings of bits to produce the overall size the image an integer multiple of an input bit length r.

The constant r itself is not to be arbitrarily taken and each hashing so far known has its own specification for the input message block length r. For example Message Digest-4 (MD4) proposed by Ronald Rivest takes r = 512 bits to produce a 128 bit hash code output. Again in the case of padding, the padding may be made by filling strings of 0-bits in to the list significant bit position of the image, or strings of bit 1 can be used. Generally, therefore, for different hash function, h, the input bit length r is different and the padding system is also different.

Processing

Once the image is padded and segmented in to t blocks each of which are r bits long, then these small blocks are fed as an input to the compression function f. Let's assume that the first block is represented by i_0, and further assume that y_0 is the result of its compression by f. Then y_0 is fed back as an input together with i_1, the next block, to produce the hash code y_1 and the procedure continues that way.
To express mathematically,

$$\left.\begin{aligned} y_0 &= f(i_0) \\ y_1 &= f(i_1, y_0) \\ y_2 &= f(i_2, y_1) \\ &\vdots \\ &\vdots \\ y_t &= f(i_t, y_{t-1}) \end{aligned}\right\} \quad (17)$$

and finally,

$$H = y_t \quad (18)$$

is the one we call the hash code or hash value of the image I and so is it can be vividly seen at this time how data integrity is achieved. Any change in any of the input blocks $i_0, i_1, i_2, \ldots$.is finally reflected in to H and hence any manipulation of the original image is simply detected and the objective fulfilled. So the design of the compression function must be in such a way that it has to fulfill the two inherent feature of a hash function criterion and the three basic properties discussed earlier.

PERFORMANCE OF HASHING FOR TEXT AND IMAGE DATA

Text data of different sizes written in MS-DOS editor and images of same sizes were hashed using the

Secure Hash Algorithm-Revised (SHA-1) [9]. SHA-1 was proposed by the National Institute for Technology and Standards (NIST) of U.S. America and accepted as a standard hashing function by the National Standards Agency (NSA). Applying this hashing function on text data and image data the following result is observed [7].

Hashing of text data and image data of the same size require the same hashing time despite the fact that characters in texts are represented by 7-bits (i.e. values from 0 to 127) and pixel elements are of 8-bits (taking values from 0 to 255) before words of 32- bit long are formed using big-endian architecture. The presence of many 0-bits in text words didn't bring performance difference, as is the case in encryption and decryption as discussed in section II(B). The very reason for this is that the operations in hashing are:

Cyclical rotations,
Logical AND-ings,
Logical OR-ings, and
Logical XOR-ings

And these operations make no distinction 0-bits and 1- bits and therefore the hashing time for texts and images of the same size are found to be identical [9].

NON-REPUDIATION THROUGH DIGITAL SIGNATURES

Privacy and data integrity are not the only issues or goals of cryptography. There is one more issue, which is known as *Non-repudiation*.

This means that an entity, particularly the sender, may deny having sent a message he originated and sent to a recipient, or may later deny a previously agreed commitment or action. The problem and solution of such cases where a third part to solve such a dispute is involved is the issue of non-repudiation.

In digital data communication on a local area network (LAN), or a globally connected Net, the Internet; it is possible to attach a sign or mark to the data which is peculiar to someone and its peculiarity to that entity be verifiable. This mark or sign is a digital data string and is known as *Digital Signature*. Digital signature serves the same purpose a hand written signature on a paper serves and it protects the recipient party from possible denial or disavowing of the sending party.

Given an m by n pixel image I and a digital signature S of the sender, the signing procedure ξ taking I and S as input produces a signed image I_s as an output is expressed mathematically as follows:

$$I_s = \xi(I,S) \tag{19}$$

Then the sender sends not I, but a signed image I_s which bears a sign which can be signed by him and only him.

Up on receiving I_s, the recipient uses a verification function ψ in any case of possible disputes to separate the image I and the signature S thereby verifying to a trusted third party that I_s contains S, which is universally known to be peculiar to the sender.

$$\psi(I_s) = I, S \tag{20}$$

Now depending on the nature of the verification function ψ, digital signatures lie under two categories. If the original message is required as input to the verification process represented by ψ, then it said to be *digital signature with appendix* [7],[9]. Again if the original message is not necessary for the verification process, and if it is further possible to recover the original message from the signature, then this type of signature scheme is said to be *digital signature with message recovery* [7],[9].

Due to its many advantages RSA Signature scheme, which has a message recovery facility, has been studied.

RSA DIGITAL SIGNATURE SCHEME

After the conceptual introduction of digital signatures by Diffie and Hellman in 1976 [2] no practical realization emerged until the work of Rivest, Shamir and Adleman in 1978. The RSA encryption and digital signature scheme is not only the first practical realization of the ideas presented in 1976 but also is the reliable and secure digital signature scheme resistant to many of the attacks so far made to break the system. That is the reason for the implementation

of RSA digital signature scheme in the Nuclear Weapon sensor monitoring station of Sandia Laboratory (U.S.) under the leadership of Gustavus J. Simmons [3].

RSA Signature Generation

For RSA Digital Signature Scheme each subscriber of a communication system who wants to enter in to a system with non-repudiation facility must generate his own private key, d, and public key pairs, (n, e), are to be generated as per the discussion in Section II(A) and the public key has to be stored in a trusted directory.

Assume subscriber P_1 wants to send a signed message to subscriber P_2. Further it is assumed in this paper that the sender is not to sign a whole message but rather the hash code of the message. This helps to minimizing signing and verification complexity and time, and has the same effect as signing the whole message because a hash code is a unique and compressed representation of message. For this reason P_1 has to attach his signature on the SHA-1 hash code output of the message.

So to sign with RSA digital signature scheme, P_1 has to perform the following procedures:

- The m by p pixel digital image I, or a text T of any size, is to be hashed using any known hashing algorithms. In our case we considered the SHA-1 hashing algorithm. The result obtained is a 160-bit hash code H of the image, or of the text.
- Using the ISO/IEC 9796 signature standard scheme shown in Fig-1, the redundant function R for H is to be computed in the range $[0, n-1]$.
- The signature S is to be computed as follows for R.

$$S = R^d \bmod n \tag{21}$$

- Then P_1's signature of H is S.

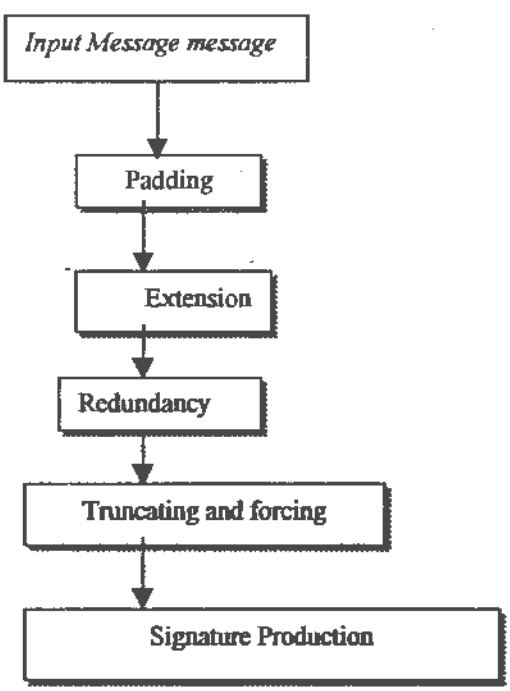


Figure 1 ISO/IEC-9796 Signature Generation Procedure

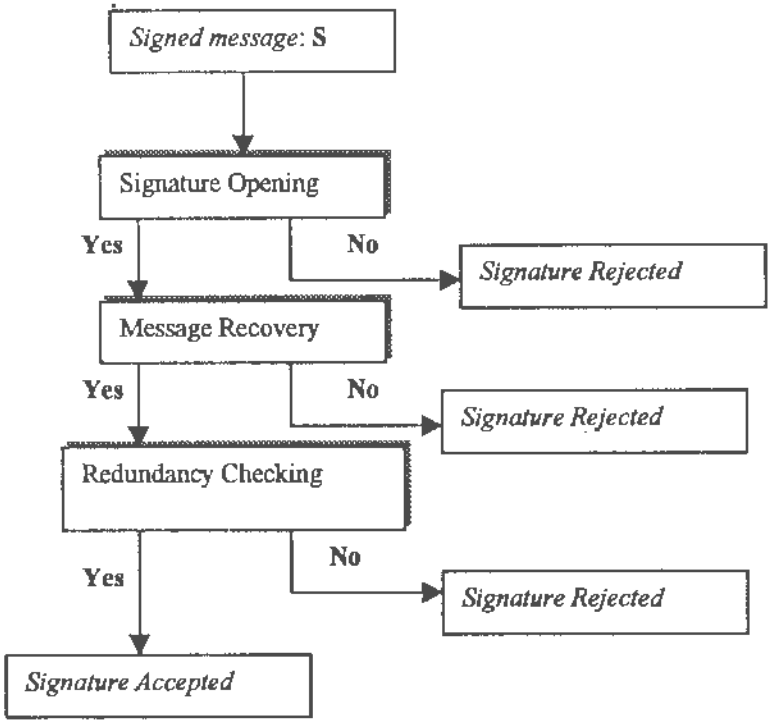


Figure 2 ISO/IEC-9796 Signature Verification Procedure

Note: The International Standardization Organization (ISO) and International Electro-technical Commission (IEC) published a standard with name ISO/IEC 9796 and standardized redundancy function and addition of redundant bits to a message before signing and in the process of verification. This standard best suits the RSA signing scheme and have been listed in [7].

RSA Signature Verification

The RSA digital scheme has the facility of recovering the message sent, in this case the hash code of the original message. Therefore, in the verification process the receiver not only verifies the authenticity of the signer, i.e. verification of the acclaimed signer, but also recovers the hash code of the message.

To Verify P_1's Signature S and recover the hash code H of the original image I, or text T, P_2 should do the following

- Get P_1's authentic public-key (n, e) from the communication system directory he trusts.
- Computes:
$$R1 = S^e \bmod n \qquad (22)$$
for each signature block received.
- Uses the ISO/IEC 9796 signature verification and message recovery facility shown in Fig. 2 to verify the signature and recover the hash code H of the original message.

PERFORMANCE OF RSA SIGNATURE SCHEME FOR TEXT AND IMAGE DATA

In section II(B) it has been stated that the encryption/ decryption performance of texts due to formation of the 32-bit words by big-endian architecture from text characters represented by 7-bits is faster than encryption/decryption of images of the same size due to formation of the 32-bit words by big-endian architecture from pixel elements represented by 8-bits [9].

In section III(B), however, this difference of bit representation is discussed for not causing any difference while hashing texts and images of the same size due to the presence of logical operations like rotation, AND-ing, OR-ing, and XOR-ing which make no difference between bits 0 and 1.

Now comes digital signature that, according to this work, assumes hashing of messages and then signing which basically means encryption and addition of redundancy bits. This means that the performance features of hashing and that of encryption are combined in digital signatures thereby slowing down the overall digital signature performance of images when compared to that of the same size of text data. This is what has been observed in this work.

CONCLUSION

The idea of encryption with RSA as a subject, the idea of data integrity with SHA-1 hashing algorithm as a subject and the idea of non-repudiation with RSA Digital Signature Scheme as a subject have been studied. As per the performances of these algorithms discussed in Sections II(B), III(B) and IV(B), secrecy systems are found to perform slower for image data types than text data types due to the discussed reasons. This enlightens some one who intends to make further research in the area and widens this result for other applications like voice communication or video communication.

Furthermore the work of this paper can be applied in areas like conditional access TV broadcasting, research areas, paper and film free hospitals, diplomatic and national security centers etc.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the Electrical Engineering Department for extending the facilities and encouragement for the production of this paper.

REFERENCES

[1] "*Encyclopedia Britannica*" William Benton Publ. 15th Ed. Vol.5, 1974.

[2] W. Diffie and M.E. Hellman: "*New Directions in Crtyptography*," IEEE Trans. Inform. Theory. Vol. It-22, pp 644 – 654, Nov. 1976.

[3] Proceedings of the IEEE Vol. 76, No-5, pp 533 – 577,May 1988.

[4] B.M. Macq and J.J. Quisquater: "*Cryptology for Digital TV Broadcasting*," Proceedings of the IEEE, Vol. 83 No-6, June 1995.

[5] Arto Salomaa: "*Public Key Cryptography*," Springer-Verlag, 2nd Ed. 1996.

[6] Vijaya Ahuja: "*Network and Internet Security*," AP-Professional 1996.

[7] Alfred J. Menzes, Paul C. van Oorschot, Scott A. Vanstone: "*Hand Book of Applied Cryptography*," CRC Press, 1997.

[8] Michael Alexander: "*Net Security: Your Digital Doberaman*," Ventanna Communication Group Inc., 1997.

[9] Getahun Mekuria: "*Cryptography for Text and Image Data Communication*," M.Sc. Thesis, Addis Ababa University, June 1999.

Appendix-A - A text message, its encrypted and decrypted forms.

The Original Text

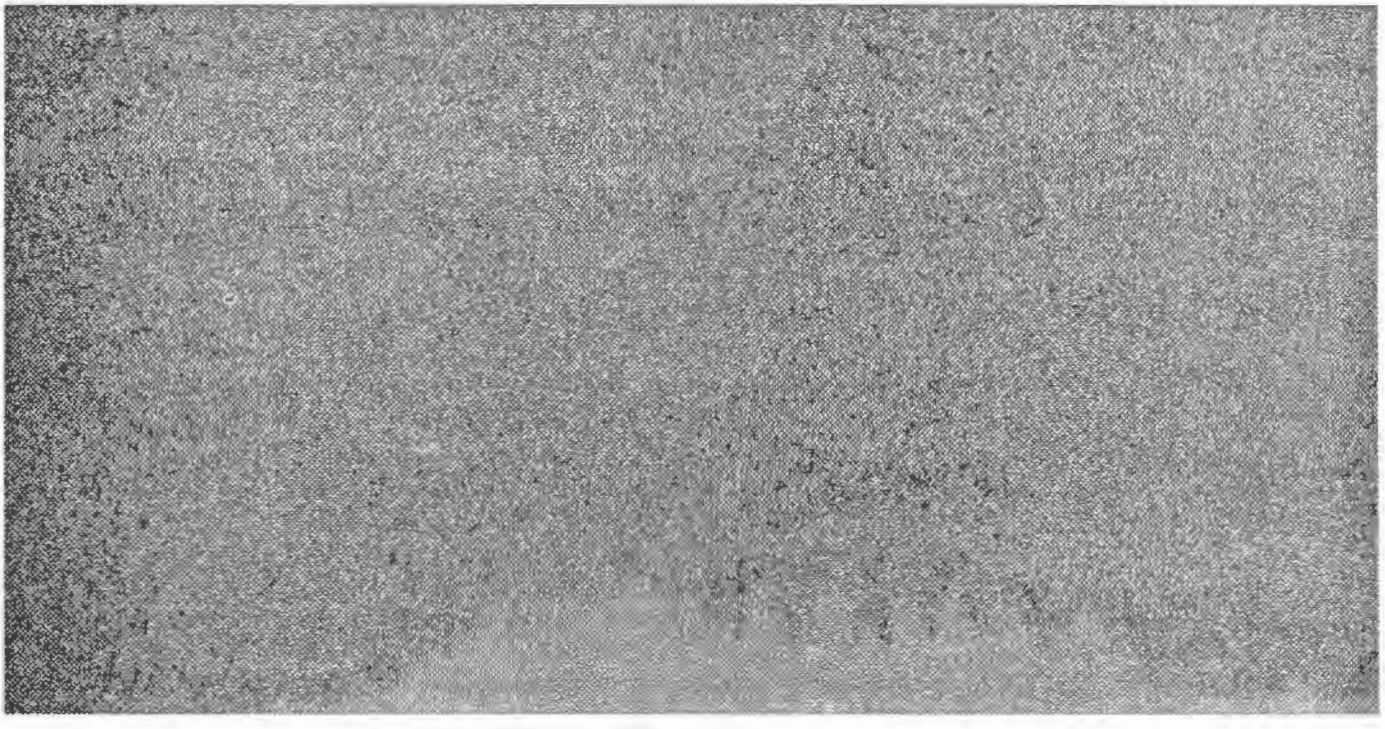
Information Secrecy deals with maintaining privacy through encryption, data integrity through hashing and non-repudiation through digital signatures; and has become the issue of today's communication engineering due to increasing failures to the secrecy of messages conveyed on a given networking system thereby creating damages beyond expectations.

The Encrypted Text

'$ñ¥ãÆ#A□Àâ□#ø¿□<ÅýÀœNÓ^—^Ê□#å™□□‡^
Yt"ÍWïØäý¡6Ç□TûÌ□yY□ÎÍ$îÚ±‡m_ÆêŒß C°□□□—Œ+ùQ¿|°'□¢$‰¡â.v¹±_R4
,ý6K5ó¯„ÜEã□¦9ÅòÑ–x□Ù□ ¡â.v
‰´Ìw□ð…QŸF□□æÎ³É >t°¸□8¾"§$—µ#®¥±Ç]4E½ƒTÑìq□‹ª¢‹X□Z□1AÙ!X±ÓmJïR□í□Àâ□°
é™ÙØ›¦□
8Ciðÿõ V□ò†^¿{$ù.3Õ)î,_q□ú,~7d'ò˜ú□8nüè□¶□`$˜gðI‰1íÓ□82AÙ!XÏki›s÷pöþþM°ý‡R□□g¶ðKkÌ□Ë <-
□□GµÍ"9£ô7□ã□-d„eæ²;
ôxÀ□ÛÚ¿ü□a³N©¬Y[›V<*E¡‡PU‰Å□ñÙ¢:ì

Appendix-B1 The Original Image

Appendix-B2 The Encrypted Image



Appendix-B3 The Decrypted Image