Secure Communication Using Encryption, Challenges and Open Issues

T. Zuva and S.M. Ngwira
Tshwane University of Technology
Computer Systems Engineering Department
Pretoria, South Africa
zuvat@tut.ac.za

ABSTRACT

Security is the fundamental concern in information systems and the data they carry that can be in either one of the three states: stationary or in transit or being processed. Institutions usually use computer systems that form part of the Internet/internet to carry their data thus making it vulnerable to attacks. In this paper we look at providing security using encryption to protect data/information. Different types of encryption schemes and cryptographic services are surveyed. Steps to create a cryptographic scheme are illustrated. Merit/demerits of the schemes, challenges and open issues in this area of cryptography are highlighted.

KEY WORDS
Encryption, Authentication, Integrity, Confidentiality, Non-repudiation

1 INTRODUCTION

Security is the fundamental concern in information systems and the data they carry that can be in either one of the three states: stationary or in transit or being processed. Institutions usually use computer systems that form part of the Internet to carry their data. The nature of Internet is that it is a shared network that makes the data in it vulnerable to malicious activities. There is need to protect data from undesirable disclosure, modification and usage. It is also necessary to protect the system from denial of service attacks and personal privacy violations. One of the mechanisms of protecting data/information is through encryption. This type of protection is often used in situations where data transmission or entity authentication is done over communications networks for which physical means of protection require unaffordable costs or difficult to implement.

Encryption is the process of converting plain text to a cryptic text to protect it from unauthorized users. Encryption is a component of cryptographic system that is defined as a set of cryptographic algorithms in combination with the key management processes that support use of the algorithms in some application context.

This paper will be arranged as follows: section 2 will deal with types of encryptions available, section 3 the functions of encryption using different types of encryptions techniques, section 4 the challenges and open issues in using encryption in security and lastly conclusion.

2 TYPES OF ENCRYPTION

There are three different types of encryption techniques that are symmetry, asymmetry and hash as shown in figure 1.

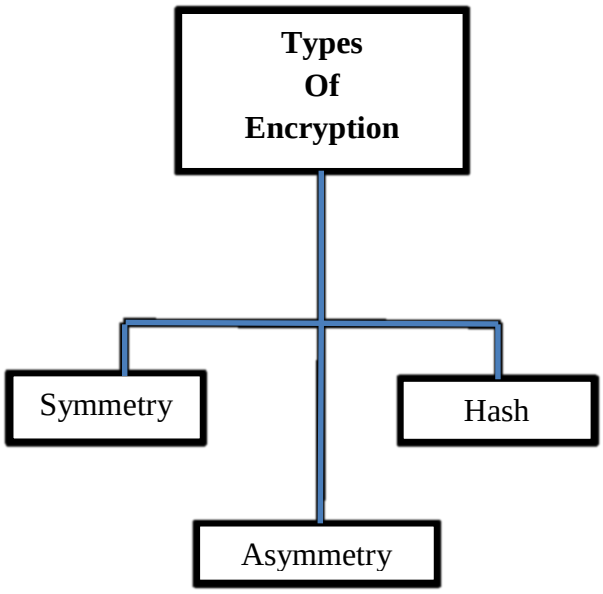

Figure 1: Types of Encryption

These are sometimes called cryptographic primitives. The description of these techniques can best derived from the following definition of cryptosystem.

Definition 1: Cryptosystem

A cryptosystem is an ordered quintet (P, C, K, E, D) fulfilling the following properties:

i. P is the finite set of Plaintexts
ii. C is the finite set of Cryptotexts
iii. K is the finite set of Keys
iv. E is the finite set of encrypting functions such that for every key $k_i \in K$ there is an encrypting function $e_{k_i} \in E : P \to C$
v. D is the finite set of decrypting functions such that for every key $k_j \in K$ there is a decrypting function $d_{k_j} \in D : C \to P$
vi. $d_{k_j}(e_{k_i}(p)) = p$ true for every Plaintext $p \in P$ and key $k_i, k_j \in K$.

The idea of key in cryptosystem is that one can only use a decrypting key if he/she has the knowledge of the encrypting key[1].

2.1 Symmetric Techniques

An encryption technique that requires the same key to encrypt a plain text and decrypt a corresponding crypto text is denoted symmetric. From definition 1 property vi, when

$$k_i = k_j \wedge d_{k_j}(e_{k_i}(p)) = p \to symmetric\ encryption$$

The following figure 2 illustrates how the symmetric techniques work.

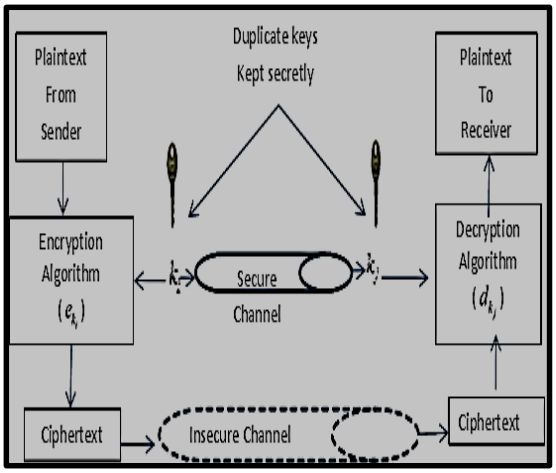

Figure 2: Illustration of Symmetric Encryption ($k_i = k_j$)

The duplicate keys k_i *and* k_j must be kept secretly by the two parties communicating as shown in figure 2. These two parties become potential source of weakness to the secure communication of the two parties[2]. Symmetric key encryption requires less processing as compared with

other encryption type that is public key encryption. It is more feasible for encrypting and decrypting large amount of information/data[3]. The implementation of symmetric encryption is done using either block or stream cipher. In block cipher block plaintext is the input while the stream cipher is usually individual characters at a time [9]. Examples of these types of symmetric key encryption are Data Encryption Standard (DES) and Translation Table for block and stream ciphers respectively. DES takes sixty-four (64-bits) plaintext block size, uses fifty-six (56-bits) key size and produces a sixty-four (64-bits) ciphertext block size. Translation Table uses usually one (1) byte (character) as an offset within one or more arrays and then the result of the translated value is written into the output stream.

The most difficult thing about symmetric encryption is the key management necessary to use them securely[4]. This entail communicating the keys to each and every person one communicates with, storing the keys, when to retire a key, etc.

2.2 Asymmetric Encryption

In asymmetric techniques the key to encrypt is different from the corresponding one used to decrypt. Property vi in the definition 1the following holds for public key encryption.

$$k_i \neq k_j \wedge d_{k_j}(e_{k_i}(p)) = p \rightarrow asymmetric\ encryption$$

This type of encryption was first invented by Diffle and Helman in 1976. The following diagram illustrates how the asymmetric encryption works.

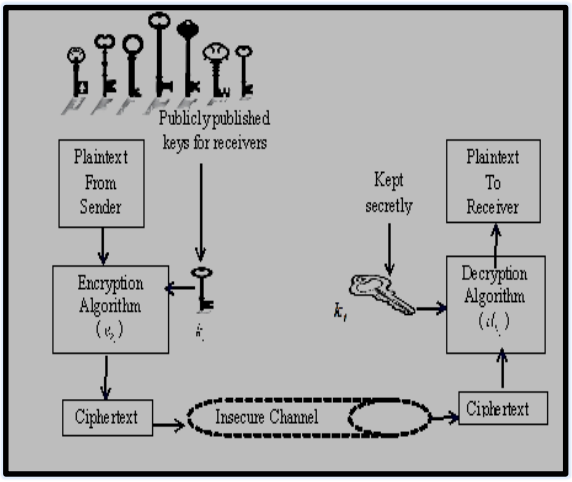

Figure 3 Illustration of Asymmetric Encryption ($k_i \neq k_j$)

Asymmetric cipher makes the public key k_i publicly available to anyone who wants to use it while at the same time keeping the private key k_j secure from the prying individuals as shown in figure 3. The private key becomes the potential source of weakness of the secure communication of the two parties. Asymmetric encryption is slower and very complicated in calculations than symmetric encryption. Due to its characteristics it is used for many other purposes other than data encryption only [5].

The implementation of asymmetric encryption done using a block of numbers derived from the plaintext as the input. Example for this type of encryption is RSA (Rivest, Shamir, and Adelman) the last names of the designers. RSA was first published in 1978. The encryption's strength depends on the choice of the key size.

2.3 Hash (message digests or one-way encryption)

Cryptographic hash is an encryption technique that does not fulfil the properties

iii, v and vi in definition 1. This requires modification of property iv as follows:

iv. E is the finite set of encrypting functions such that $h \in E : P \rightarrow C$

A hash is a fixed length word that precisely describes a message. The following figure 4 illustrates how hash encryption works.

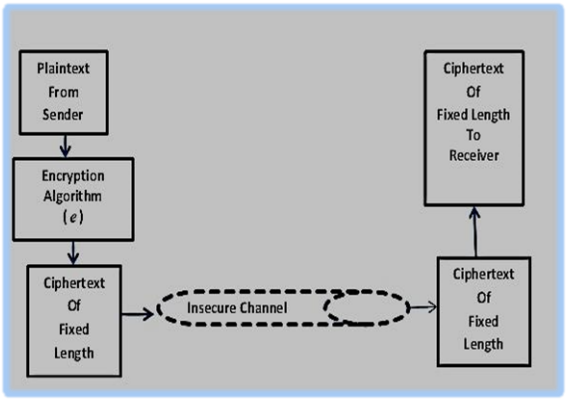

Figure 4: Illustration of hash encryption

Hash functions, also called message digests and one-way encryption, are algorithms that use no key. Instead, a fixed-length hash value is computed based upon the plaintext that makes it impossible for either the contents or length of the plaintext to be recovered[6, 7]. Hash algorithms are typically used to provide a digital fingerprint of a file's contents often used to ensure that the file has not been altered by an intruder or virus. Hash functions are also commonly employed by many operating systems to encrypt passwords. Example of hash encryption is Message Digest 5 (MD5). MD5 produces a one hundred and twenty eight (128)-bits fixed length for every given plaintext but with a different value of the hash[8].

3 CRYPTOGRAPHIC SERVICES

Cryptography is used to achieve security goals. An application may require that either all or some of the goal are achieved at the same time. The main goals of cryptography are listed below and shown in figure 5.

- Authentication
- Data integrity
- Confidentiality
- Non-repudiation

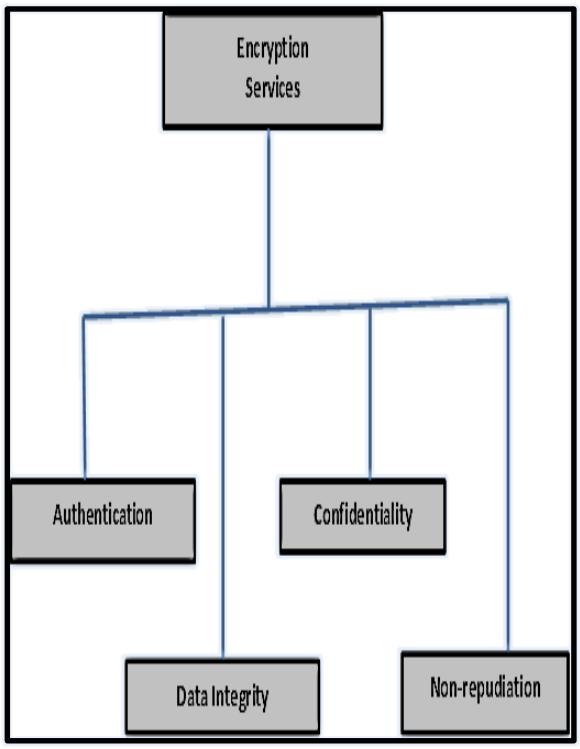

Fig 5: Cryptographic Services

Authentication is a service that ensures that the information/data originated from a particular party and that it still maintains the integrity. This service verifies the identity of the user or system that created the information/data and that the information/data have not been modified. Data integrity entails ensuring that information/data have not been changed since its creation, transmission and/or storage. Secrecy goes hand-in-hand with

confidentiality where the information/data have not been disclosed to unauthorized parties. Non-repudiation is a service that the sending party cannot deny sending the information/data. This service provides assurance of integrity and origins of information/data to the receiver.

Fulfilment of these services can be done through a combination of cryptographic primitives to create a cryptographic scheme. Thus users and developers are faced with a range of choices in their use of cryptographic mechanisms [9].

Designers of secure systems often begin by considering which security services are needed to protect the information contained within and processed by the system. After these services have been determined, the designer then considers what mechanisms will best provide these services[10]. Cryptographic mechanisms consisting of algorithms, keys, and other keying material often provide the most cost-effective means of protecting the security of information where the information would otherwise be exposed to unauthorized entities. In many cases different algorithms need to be employed in order to provide all the desired services.

For instance there is a requirement of data integrity, privacy and confidentiality, non-repudiation and user authentication then it necessary to come out with a combination of the cryptographic primitives. A set of cryptographic primitives needs to be identified to provide the required security services. Table 1 shows the primitives that we have identified to provide the services.

Table 1: A set of cryptographic Primitives identified

Cryptographic Primitives	Purpose	Service
Symmetric encryption	Encrypting messages	Privacy and Confidentiality
Hash function	Ensure contents of a message have not been tempered with to the receiver	Data integrity
Asymmetric encryption	Ensure the exchange of the secret key is achieved securely	non-repudiation and user authentication

After identifying the set of primitives to use, the combination of these cryptographic primitives to make a security protocol then follows. There are so many ways of combining them and this requires expertise to find the optimal combination. There is very little formal analysis of protocols to ensure that the chosen combination is the optimal one[11]. There are methods for full verification but they are extremely cumbersome and cannot be automated foe example BAN logic, The Strand Spaces Approach[12].

4 CHALLENGES AND OPEN ISSUES

The challenges of providing the cryptographic services are to identify the cryptographic services required, to decide whether to use cryptographic algorithms that are in existence already or to design new algorithms and how best to optimize the use of cryptographic algorithms to provide the determined services. The key management is another issue that needs to be attended to. Security using cryptography without proper management of the keys implies that the encryption security is not worth it. The keys need to be managed from inception up to

retirement. The length of the keys to be used must be determined depending on the usefulness of the information being protected. All the mentioned above constitute the open issues being investigated on.

5 CONCLUSION

A brief survey was conducted on cryptography and the basic algorithms explained. The cryptographic services were looked at. The challenges and open issues of providing cryptographic services were discussed and the importance of key management. The significance of key management cannot be over emphasized due to the factor that there is no security if the keys used in cryptography are not managed well. The length of the keys must depend on the usefulness and importance of the information being protected even the devises being used. Some devices have problems with memory thus it not necessary to consume too much memory due to key storage.

REFERENCES

[1] K.Govinda and E.Sathiyamoorthy, "Identity Anonymization and Secure Data Storage using Group Signature in Private Cloud," *IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS),*, vol. 1, pp. 115-118, 2011.

[2] D. S. Abdul, E. H. M., A. Kader, and M. M. Hadhoud, "Performance Evaluation of Symmetric Encryption Algorithms," *Communications of the IBIMA*, vol. 8, pp. 58-64, 2009.

[3] R. Masram, V. Shahare, J. Abraham, and R. Moona, "ANALYSIS AND COMPARISON OF SYMMETRIC KEY CRYPTOGRAPHIC ALGORITHMS BASED ON VARIOUS FILE FEATURES," *International Journal of Network Security & Its Applications (IJNSA)*, vol. 6, pp. 43-52, 2014.

[4] R. Masram, V. Shahare, J. Abraham, and R. Moona, "ANALYSIS AND COMPARISON OF SYMMETRIC KEY CRYPTOGRAPHIC ALGORITHMS BASED ON VARIOUS FILE FEATURES," *International Journal of Network Security & Its Applications (IJNSA), Vol.6, No.4, July 2014*, vol. 6, pp. 43-54, 2014.

[5] S. FARAH, M. Y. JAVED, A. SHAMIM, and T. NAWAZ, "An experimental study on Performance Evaluation of Asymmetric Encryption Algorithms," in *Recent Advances in Information Science*, Paris, France, 2012, pp. 121-126.

[6] R. P. (Arya), U. Mishra, and A. Bansal, "A Survey on Recent Cryptographic Hash Function Designs," *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, vol. 2, pp. 117-122, 2013.

[7] R. Watro, D. Kong, S.-f. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: securing sensor networks with public key technology," in *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, 2004, pp. 59-64.

[8] I. Mironov, "Hash functions: Theory, attacks, and applications," *Microsoft Research, Silicon Valley Campus. Noviembre de*, 2005.

[9] S. Zhu, S. Setia, and S. Jajodia, "LEAP+: Efficient security mechanisms for large-scale distributed sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, pp. 500-528, 2006.

[10] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Efficient Security Mechanisms for Routing Protocolsa," in *NDSS*, 2003.

[11] S. Lal, M. Jain, and V. Chaplot, "Approaches to Formal Verification of Security Protocols," *CoRR*, vol. abs/1101.1815, pp. 1-9, 2011.

[12] M. Avalle, A. Pironti, and R. Sisto, "Formal verification of security protocol implementations: a survey " *Formal Aspects of Computing* vol. 26, pp. 99-123, January 2014.