Encryption: Comparison of Security Standards

Brandon Rowe

USC Upstate Senior  
800 University Way,   
Spartanburg, SC 29303  
864-256-1502

bhrowe@email.uscupstate.edu

**ABSTRACT**

Encryption is the process of scrambling a message so that only the intended recipient can read it. Encryption can provide a means of securing information. As more and more information is stored on computers or communicated via computers, the need to insure that this information is invulnerable to snooping and/or tampering becomes more relevant. With the fast progression of digital data exchange in electronic way, Information Security is becoming much more important in data storage and transmission. Information Confidentiality has a prominent significance in the study of ethics, law and most recently in Information Systems. With the evolution of human intelligence, the art of cryptography has become more complex in order to make information more secure. Arrays of Encryption systems are being deployed in the world of Information Systems by various organizations. In this paper, a survey of various Encryption Algorithms is presented.

**Categories and Subject Descriptors**

D.3.3 [**Programming Languages**]: Language Contructs and Features – *abstract data types, polymorphism, control structures.* This is just an example, please use the correct category and subject descriptors for your submission*.* The ACM Computing Classification Scheme: <http://www.acm.org/class/1998/>

**General Terms**

Information Security, Encryption

**Keywords**

Encryption, SHA-1, SHA-3, SHA-256, RSA, DES, AES

# INTRODUCTION

In recent years, a lot of applications based on internet have emerged such as on-line shopping, stock trading, internet banking and electronic bill payment etc. Such transactions, over wire or wireless public networks demand end-to-end secure connections, should be confidential, to ensure data authentication, accountability and confidentiality, integrity and availability, also known as CIA triad [1].

The NIST Computer Security Handbook [NIST95] defines the term computer security as, “The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability and confidentiality of information system resources (includes hardware, software, firmware, information/data, and telecommunications).” Security is the mechanism by which information and services are protected from unintended or unauthorized access, change or destruction. Security in networking is based on Cryptography (a word with Greek origins, means “secret writing”), the science and art of transforming messages to make them secure and immune to attack [2].

Encryption is one of the principal means to guarantee security of sensitive information. Encryption algorithm performs various substitutions and transformations on the plaintext (original message before encryption) and transforms it into ciphertext (scrambled message after encryption). Many encryption algorithms are widely available and used in information security. Encryption algorithms are classified into two groups: Symmetric-key (also called secret-key) and Asymmetric-key (also called public-key) encryption.

Symmetric key encryption is a form of cryptosystem in which encryption and decryption are performed using the same key. It is also known as conventional encryption.

Asymmetric encryption is a form of cryptosystem in which encryption and decryption are performed using the different keys – one a public key and one a private key. It is also known as public-key encryption [3].

A Key is a numeric or alpha numeric text or may be a special symbol. The Key is used at the time of encryption takes place on the Plain Text and at the time of decryption takes place on the Cipher Text. The selection of key in Cryptography is very important since the security of encryption algorithm depends directly on it. The strength of the encryption algorithm relies on the secrecy of the key, length of the key, the initialization vector, and how they all work together [4].

Asymmetric encryption techniques are about 1000 times slower than Symmetric encryption which makes it impractical when trying to encrypt large amounts of data. Also to get the same security strength as symmetric, asymmetric must use a stronger key than symmetric encryption technique [5]. The classification of major encryption techniques is shown in Figure 1.

# RELATED Work

To give more perspective about the performance of the encryption algorithms, this subsection describes and examines previous work done in field of data encryption. The metrics taken into consideration are processing speed, throughput, power consumption, avalanche effect, packet size and data types. This subsection also discusses the results obtained for some of the algorithms.

Arora et al. [6] studied about the performance of different security algorithms on a cloud network and also on a single processor for different input sizes. This paper aims to find in quantitative terms like Speed-Up Ratio that benefits of using cloud resources for implementing security algorithms (RSA, MD5 and AES) which are used by businesses to encrypt large volumes of data. Three different kinds of algorithms are used – RSA (an asymmetric encryption algorithm), MD5 (a hashing algorithm) and AES (a symmetric encryption algorithm).

The results reported in this paper conclude that the algorithms implemented on cloud environment (i.e. Google App) are more efficient than using them on single system. For both uni- processor (local) as well as cloud (Appengine) environment, RSA is the most time consuming and MD5 is the least. Highest Speed-Up Ratio is obtained in AES for low input file sizes and the Speed-Up Ratio falls sharply as the input file size is increased. For each input size, the Speed-Up Ratio is highest for AES, followed by MD5 and least for RSA algorithm.

Seth et al. [7] have done the comparative analysis of three algorithms; RSA, DES and AES while considering certain parameters such as computation time, memory usage and output byte. These parameters are the major issue of concern in any Encryption Algorithm. Experimental results show that DES algorithm consumes least encryption time and AES algorithm has least memory usage while encryption time difference is very minor in case of AES and DES algorithm. RSA consume longest encryption time and memory usage is also very high but output byte is least in case of RSA algorithm.

Abdul. Elminaam et al. [5] studied about the performance of Symmetric Encryption Algorithms. This paper provides evaluation of six of the most common encryption algorithms: AES (Rijndael), DES, 3DES, RC2, Blowfish, and RC6. A comparison has been conducted at different settings for each algorithm such as different sizes of data blocks, different data types, battery power consumption, different key size and finally encryption/decryption speed. Experimental simulation shows following results. There is no significant difference when the results are displayed either in hexadecimal base encoding or in base 64 encoding. In case of changing packet size, it was found that RC6 requires less time than all algorithms except Blowfish. In case of changing data type such as image instead of text, it was found that RC2, RC6 and Blowfish has disadvantage over other algorithms in terms of time consumption. Also, 3DES still has low performance compared to algorithm DES. Finally -in the case of changing key size (possible only in AES and RC6 algorithms) it can be seen that higher key size leads to clear change in the battery and time consumption.

Pavithra et al. [8] compares the performance evaluation of various cryptographic algorithms. On the basis of parameter taken as time various cryptographic algorithms are evaluated

# Encryption Algorithms

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*Conference’04*, Month 1–2, 2004, City, State, Country.

Copyright 2004 ACM 1-58113-000-0/00/0004…$5.00.

## Encryption is a well-known technology for protecting sensitive data. Use of the combination of Public and Private Key encryption to hide the sensitive data of users, and cipher text retrieval [6].

## a) *Data Encryption Standard (DES)*

## DES (Data Encryption Standard) algorithm

## purpose is to provide a standard method for protecting sensitive commercial and unclassified data. In this same key used for encryption and decryption process [7]. DES algorithm consists of the following steps

## i. Encryption

## 1. DES accepts an input of 64-bit long plaintext and 56-bitkey (8 bits of parity) and produce output of 64 bit block.

## 2. The plaintext block has to shift the bits around.

## 3. The 8 parity bits are removed from the key by subjecting the key to its Key Permutation.

## 4. The plaintext and key will processed by following

## i. The key is split into two 28 halves

## ii. Each half of the key is shifted (rotated) by one or two bits, depending on the round.

## iii. The halves are recombined and subject to a compression permutation to reduce the key from 56 bits to 48 bits. This compressed keys used to encrypt this round’s plaintext block.

## iv. The rotated key halves from step 2 are used in next round.

## v. The data block is split into two 32-bit halves.

## vi. One half is subject to an expansion permutation to increase its size to 48 bits.

## vii. Output of step 6 is exclusive-OR’ed with the 48-itcompressed key from step 3.

## viii. Output of step 7 is fed into an S-box, which substitutes key bits and reduces the 48-bit block back down to 32-bits.

## ix. Output of step 8 is subject to a P-box to permute the bits.

x. The output from the P-box is exclusive-OR’ed with other half of the data block. k. The two data halves are swapped and become the next round’s input.

## b) *Advanced Encryption Standard (AES)*

Advanced Encryption Standard (AES) algorithm is not only for security but also for great speed. Both hardware and software implementation are faster still. New encryption standard recommended by NIST to replace DES. Encrypts data blocks of 128 bits in 10, 12 and 14 round depending on key size as shown in Figure - 2. It can be implemented on various platforms specially in small devices. It is carefully tested for many security applications.

## i. *Algorithm Steps* : These steps are used to encrypt 128-bit block

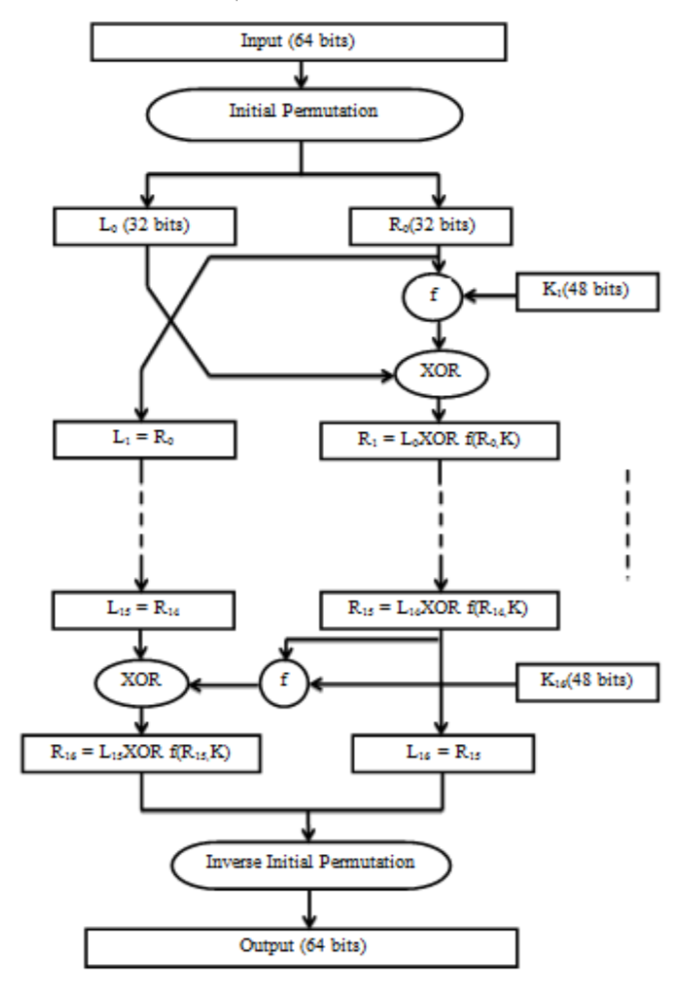


Figure 1. Diagram of DES Algorithm

.

## 1. Get the set of round keys from the cipher key.

## 2. Initialize state array and add the initial round key to the starting state array.

## 3. Perform round = 1 to 9 : Execute Usual Round.

## 4. Execute Final Round.

5. Corresponding cipher text chunk output of Final Round Step.

## ii. *Usual Round* : Execute the following operations which are described above.

## 1. Sub Bytes

## 2. Shift Rows

## 3. Mix Columns

## 4. Add Round Key, using K(round)

## iii. *Final Round* : Execute the following operations which are described above.

## 1. Sub Bytes

## 2. Shift Rows

## 3. Add Round Key, using K(10)

## iv. *Encryption* : Each round consists of the following four steps:

## 1. Sub Bytes : The first transformation is used at the encryption site. To substitute a byte, we interpret the byte as two hexadecimal digits.

## 2. Shift Rows : In the encryption, the transformation is called Shift Rows.

## 3. Mix Columns : The Mix Columns transformation operates at the column level; it transforms each column of the state to a new column.

4. Add Round Key : Add Round Key proceeds one column at a time. Add Round Key adds a round key word with each state column matrix the operation in Add Round Key is matrix addition.

5. The last step consists of XO Ring the output of the previous three steps with four words from the key schedule. The last round for encryption does not involve the “Mix columns” step. [8]

## a) *Rivest-Shamir-Adleman (RSA)*

## RSA is a widely used Public-Key algorithm. RSA was first invented in 1977. In our proposed work, we are using RSA algorithm to encrypt the data to provide security so that only the concerned user can access it.

RSA algorithm involves these steps:

## 1. Key Generation

## 2. Encryption

## 3. Decryption

## i. Key Generation

Before the data is encrypted, Key generation should be done. [9]

Steps:

Generate a public/private key pair:

1. Generate two large distinct primes p and q
2. Compute n = pq and φ = (p − 1)(q − 1)
3. 3. Select an e, 1 < e < φ, relatively prime to φ.
4. 4. Compute the unique integer d, 1 < d < φ where ed ≡φ 1.
5. Return public key (n, e) and private key d
   1. Encryption

Encryption is the process of converting original plain text (data) into cipher text (data).

Encryption with key (n, e)

1. Represent the message as an integer m € {0, . . . , n − 1}
2. Compute c = msube mod n

## Title and Authors

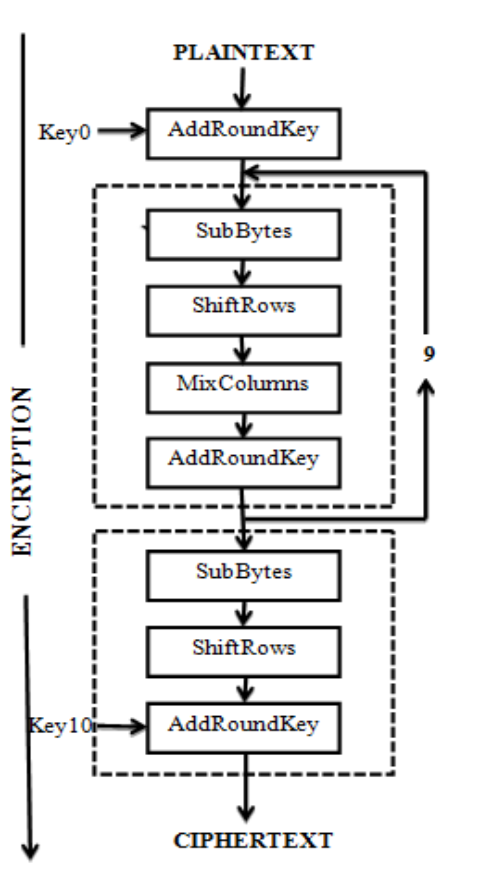


Figure 1. Diagram of AES Algorithm

.

The title (Helvetica 18-point bold), authors' names (Helvetica 12-point) and affiliations (Helvetica 10-point) run across the full width of the page – one column wide. We also recommend phone number (Helvetica 10-point) and e-mail address (Helvetica 12-point). See the top of this page for three addresses. If only one address is needed, center all address text. For two addresses, use two centered tabs, and so on. For more than three authors, you may have to improvise.[[1]](#footnote-1)

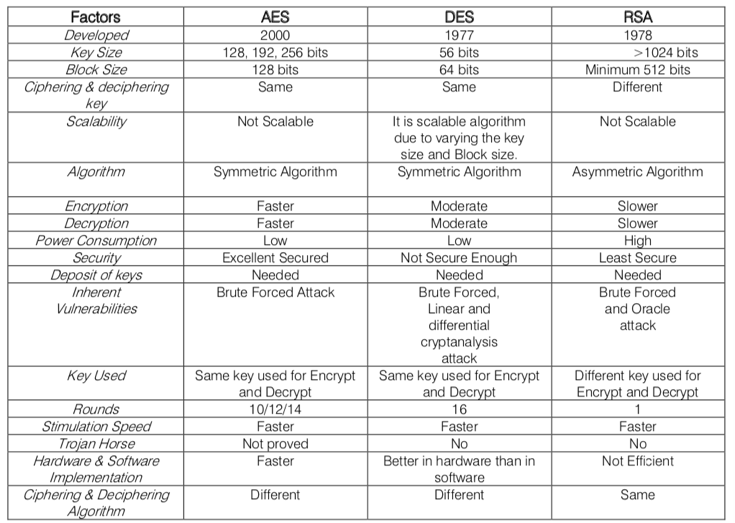
## First Page Copyright Notice

Please leave 3.81 cm (1.5") of blank text box at the bottom of the left column of the first page for the copyright notice.

## Subsequent Pages

For pages other than the first page, start at the top of the page, and continue in double-column format. The two columns on the last page should be as close to equal length as possible.

Table 1. Comparison between AES, DES & RSA



## References and Citations

Footnotes should be Times New Roman 9-point, and justified to the full width of the column.

Use the standard Communications of the ACM format for references – that is, a numbered list at the end of the article, ordered alphabetically by first author, and referenced by numbers in brackets [1]. See the examples of citations at the end of this document. Within this template file, use the style named references for the text of your citation.

The references are also in 9 pt., but that section (see Section 7) is ragged right. References should be published materials accessible to the public. Internal technical reports may be cited only if they are easily accessible (i.e. you can give the address to obtain the report within your citation) and may be obtained by any reader. Proprietary information may not be cited. Private communications should be acknowledged, not referenced (e.g., “[Robertson, personal communication]”).

## Page Numbering, Headers and Footers

Do not include headers, footers or page numbers in your submission. These will be added when the publications are assembled.

# FIGURES/CAPTIONS

Place Tables/Figures/Images in text as close to the reference as possible (see Figure 1). It may extend across both columns to a maximum width of 17.78 cm (7”).

Captions should be Times New Roman 9-point bold. They should be numbered (e.g., “Table 1” or “Figure 2”), please note that the word for Table and Figure are spelled out. Figure’s captions should be centered beneath the image or picture, and Table captions should be centered above the table body.

# SECTIONS

The heading of a section should be in Times New Roman 12-point bold in all-capitals flush left with an additional 6-points of white space above the section head. Sections and subsequent sub- sections should be numbered and flush left. For a section head and a subsection head together (such as Section 3 and subsection 3.1), use no additional space above the subsection head.

## Subsections

The heading of subsections should be in Times New Roman 12-point bold with only the initial letters capitalized. (Note: For subsections and subsubsections, a word like *the* or *a* is not capitalized unless it is the first word of the header.)

### Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized and 6-points of white space above the subsubsection head.

#### Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized.

#### Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized.

# ACKNOWLEDGMENTS

Our thanks to ACM SIGCHI for allowing us to modify templates they had developed.

# REFERENCES

1. Bowman, B., Debray, S. K., and Peterson, L. L. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst., 15,* 5 (Nov. 1993), 795-825.
2. Ding, W., and Marchionini, G. *A Study on Video Browsing Strategies.* Technical Report UMIACS-TR-97-40, University of Maryland, College Park, MD, 1997.
3. Fröhlich, B. and Plate, J. The cubic mouse: a new device for three-dimensional iput. In *Proceedings of the SIGCHI conference on Human factors in computing systems   
   (CHI ’00)* (The Hague, The Netherlands, April 1-6, 2000). ACM Press, New York, NY, 2000, 526-531.
4. Lamport, L. *LaTeX User’s Guide and Document Reference Manual.* Addison-Wesley, Reading, MA, 1986.
5. Sannella, M. J. *Constraint Satisfaction and Debugging for Interactive User Interfaces.* Ph.D. Thesis, University of Washington, Seattle, WA, 1994.

Columns on Last Page Should Be Made As Close As Possible to Equal Length

Encryption: Tunnel to the Future

SHA – 1 (AKA SHA-0)

Encryption should encompass every facet of computer, network and data science in the 21st century. Encryption should be the foundation of every transaction, connection or software established in the future. Encryption creates a digital tunnel to help secure individuals in the digital age.

In this paper, we describe the past, present and future developments in encryption as a security mechanism. We discuss why encryption is important and where the areas of greatest focus may be applied for future encryption algorithms and tools. To understand the research we must provide background of each development in security algorithms over time while also supplying times and dates at which the algorithms were broken or even if they have been broken to date. We will discuss outdated encryption algorithms as well as current encryption standards. The culmination of this project will be an understanding of encryption to date and the potential for encryption standards in the future.

In cryptography, SHA-1 (Secure Hash Algorithm 1) is a cryptographic hash function which takes an input and produces a 160-bit (20-byte) hash value known as a message digest – typically rendered as a hexadecimal number, 40 digits long. It was designed by the United States National Security Agency, and is a U.S. Federal Information Processing Standard.

Since 2005 SHA-1 has not been considered secure against well-funded opponents, and since 2010 many organizations have recommended its replacement by SHA-2 or SHA-3.

All major web browser vendors ceased acceptance of SHA-1 SSL certificates in 2017. In February 2017, CWI Amsterdam and Google announced they had performed a collision attack against SHA-1, publishing two dissimilar PDF files which produced the same SHA-1 hash.

SHA-1 produces a message digest based on principles similar to those used by Ronald L. Rivest of MIT in the design of the MD2, MD4 and MD5 message digest algorithms, but generates a larger hash value (160 bits vs. 128 bits).

SHA-1 forms part of several widely used security applications and protocols, including TLS and SSL, PGP, SSH, S/MIME, and IPsec. Those applications can also use MD5; both MD5 and SHA-1 are descended from MD4.

SHA-1 and SHA-2 are the hash algorithms required by law for use in certain U.S. government applications, including use within other cryptographic algorithms and protocols, for the protection of sensitive unclassified information. FIPS PUB 180-1 also encouraged adoption and use of SHA-1 by private and commercial organizations. SHA-1 is being retired from most government uses; the U.S. National Institute of Standards and Technology said, "Federal agencies should stop using SHA-1 for...applications that require collision resistance as soon as practical, and must use the SHA-2 family of hash functions for these applications after 2010" (emphasis in original), though that was later relaxed to allow SHA-1 to be used for verifying old digital signatures and time stamps.

A prime motivation for the publication of the Secure Hash Algorithm was the Digital Signature Standard, in which it is incorporated.

The SHA hash functions have been used for the basis of the SHACAL block ciphers.

Analysis and validation

For a hash function for which L is the number of bits in the message digest, finding a message that corresponds to a given message digest can always be done using a brute force search in approximately 2L evaluations. This is called a preimage attack and may or may not be practical depending on L and the particular computing environment. However, a collision, consisting of finding two different messages that produce the same message digest, requires on average only about 1.2 × 2L/2 evaluations using a birthday attack. Thus the strength of a hash function is usually compared to a symmetric cipher of half the message digest length. SHA-1, which has a 160-bit message digest, was originally thought to have 80-bit strength.

In 2005, cryptographers Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu produced collision pairs for SHA-0 and have found algorithms that should produce SHA-1 collisions in far fewer than the originally expected 280 evaluations.

Some of the applications that use cryptographic hashes, like password storage, are only minimally affected by a collision attack. Constructing a password that works for a given account requires a preimage attack, as well as access to the hash of the original password, which may or may not be trivial. Reversing password encryption (e.g. to obtain a password to try against a user's account elsewhere) is not made possible by the attacks. (However, even a secure password hash can't prevent brute-force attacks on weak passwords.)

In the case of document signing, an attacker could not simply fake a signature from an existing document: The attacker would have to produce a pair of documents, one innocuous and one damaging, and get the private key holder to sign the innocuous document. There are practical circumstances in which this is possible; until the end of 2008, it was possible to create forged SSL certificates using an MD5 collision.

Due to the block and iterative structure of the algorithms and the absence of additional final steps, all SHA functions (except SHA-3[23]) are vulnerable to length-extension and partial-message collision attacks.[24] These attacks allow an attacker to forge a message signed only by a keyed hash—SHA(message || key) or SHA(key || message)—by extending the message and recalculating the hash without knowing the key. A simple improvement to prevent these attacks is to hash twice: SHAd(message) = SHA(SHA(0b || message)) (the length of 0b, zero block, is equal to the block size of the hash function).

AES

1. If necessary, you may place some address information in a footnote, or in a named section at the end of your paper. [↑](#footnote-ref-1)