Implementation Approach in JAVA for Hybrid (Multi-Level) Crypto Systems

Sivananda Lahari Reddy. Elicherla[1] Dr. Raghunatha Reddy. Vandavagula[2]
[1]Research Scholar [2]Assistant Professor
[1,2]Department of Computer Science & Technology
[1,2]S.K.University, Anantapur, A.P. -515003, India

Abstract— Hybrid encryption is a method of encryption that integrates two or more encryption systems. It combines a blend of symmetric, asymmetric and hashing encryption to gain from the strength of each form of encryption. The objective of this article is to summarize the implementation approach of all hybrid crypto systems namely symmetric hashing, asymmetric hashing and multi-level hashing crypto systems using JAVA. These cryptographic algorithms are being evaluated to devise a potential robust cryptographic system which provides more security and makes the encryption further tough compared to the existing hybrid algorithms to protect the data integrity and maintain the confidentiality.

Keywords: Cryptography, Hybrid Encryption, Security, Symmetric, Asymmetric, Hashing Algorithms, Hybrid Cryptography, Data Integrity and confidentiality

I. INTRODUCTION

At present, insecure data is on the raise used in communication over the internet. As a result security of data is a foremost interest of internet users. The best solution is to use some of the cryptographic algorithms which encrypt the data in cipher format and transmit it over the Net and again decrypted to unadulterated data. The field of cryptography deals with the strategy for transmitting the information securely. The purpose is to allow the intended recipients of a message to receive the message properly while disrupting the unintended users from understanding the message.

Cryptography is a popular way of sending vital information secretly. It includes techniques derived from mathematical concepts and a set of rule-based calculations called algorithms to transform messages in ways that are hard to decipher. Therefore data security has become one of the most necessary factors that need attention during the process of data transfer.

The significance of information and communication systems for society and for global market is escalating with the growing value and quantity of data that is transmitted and stored on those systems.

The hiding of information is called encryption, and when the information is unhidden, it is called decryption. A cipher is used to achieve the encryption and decryption. The information that is being hidden is called plaintext; once it has been encrypted, it is called cipher text.

Hybrid encryption is a mode of encryption that merges two or more encryption systems. It incorporates a combination of asymmetric, asymmetric and hashing encryption to benefit from the strengths of each form of encryption.

This journal presents assessment various cryptographic systems such as symmetric, asymmetric, and hashing and combination of symmetric, asymmetric and hashing algorithms to discover the probable cryptographic solutions that make the encryption process more robust, secure and cumbersome.

This piece of work is being performed as a part of the research work - "Data Security Optimization through Hybrid Cryptography Using Symmetric (AES), Asymmetric (RSA) & Hashing (MD5) Algorithms"

II. CRYPTOGRAPHY

Cryptography is a method of protecting information and communications through the use of codes, so that only those for whom the information is intended can read and process it. The prefix "crypt-" means "hidden" or "vault" -- and the suffix "-graphy" stands for "writing.

Cryptography uses two types of keys: symmetric and asymmetric. Symmetric keys have been found the longest; they make use of a single key for both the encryption and decryption of the cipher text. This type of key is called a secret key. Most cryptographic processes use symmetric encryption to encrypt data transmission but use asymmetric encryption to encrypt and exchange the secret key. Symmetric encryption, also known as private key encryption, uses the same private key for both encryption and decryption. Whereas Asymmetric Encryption uses two distinct, yet related keys. One key, the Public Key, is used for encryption and the other, the Private Key, is for decryption. As implied in the name, the Private Key is intended to be private so that only the authenticated recipient can decrypt the message.

As this research prime objective is to compare the effectiveness of hybrid cryptosystems, it's important to understand the symmetric, asymmetric and hashing algorithms in brief and also to know the approach followed in implementing these crypto systems.

A. Symmetric Key Cryptography

Symmetric-key cryptography is also known as private-key cryptography; a secret key may be held by one person or exchanged between the sender and the receiver of a message. If private key cryptography is used to send secret messages between two parties, both the sender and receiver must have a copy of the secret key.

B. Asymmetric Key Cryptography

In the two-key system is also known as the public key system, one key encrypts the information and another, mathematically related key decrypts it. The computer sending an encrypted message uses a chosen private key that is never shared and so is known only to the sender. If a sending computer first encrypts the message with the intended receiver's public key and again with the sender's secret, private key, then the receiving computer may decrypt

the message, first using its secret key and then the sender's public key. Using this public-key cryptographic method, the sender and receiver can authenticate one another as well as protect the secrecy of the message.

C. Cryptographic Hash Function (CHF)

A cryptographic hash function (CHF) is a hash function that is suitable for use in cryptography. It is a mathematical algorithm that maps data of arbitrary size (often called the "message") to a bit string of a fixed size (the "hash value", "hash", or "message digest") and is a one-way function, that is, a function which is practically infeasible to invert. The input to the hash function is of arbitrary length but the output is always of fixed length.

III. Cryptographic Algorithms

There are three algorithms considered for the research as follows to discover the potential robust crypto system by combining the strengths of each algorithm.

A. Advanced Encryption Standard (AES)

The Advanced Encryption Standard, AES, is a symmetric encryption algorithm and one of the most secure. This method uses a block cipher, which encrypts data one fixed-size block at a time, unlike other types of encryption, such as stream ciphers, which encrypt data bit by bit. AES is comprised of AES-128, AES-192, and AES-256.

Since AES is symmetric key encryption, you must share the key with other individuals for them to access the encrypted data. Furthermore, if you don't have a secure way to share that key and unauthorized individuals gain access to it, they can decrypt everything encrypted with that specific key. It is a symmetric-key algorithm, meaning each recipient must receive the key through a different channel than the message.

B. Rivest-Shamir-Adleman (RSA)

This asymmetric algorithm is named after Ron Rivest, Adi Shamir, and Len Adelman. It uses public-key cryptography to share data over an insecure network. There are two keys: one public and one private. The public key is just as the name suggests: public. Anyone can access it. However, the private key must be confidential. When using RSA cryptography, you need both keys to encrypt and decrypt a message. You use one key to encrypt your data and the other to decrypt it.

According to Search Security, RSA is secure because it factors large integers that are the product of two large prime numbers. Additionally, the key size is large, which increases security. Most RSA keys are 1024-bits and 2048-bits long. However, the longer key size does mean it's slower than other encryption methods.

C. Message-Digest 5 (MD5)

In cryptography, MD5 (Message-Digest algorithm 5) is a widely used cryptographic hash function with a 128-bit hash value. As an Internet standard (RFC 1321), MD5 has been employed in a wide variety of security applications and is also commonly used to check the integrity of files. An MD5 hash is typically expressed as a 32 digit hexadecimal number.

IV. Implementation Approach

As the research major objective is to optimize the data security using hybrid crypto systems, the following approach has been followed to implement all crypto systems namely symmetric, asymmetric and multi-level hybrid crypto systems using JAVA.

A. Advanced Encryption Standard (AES)

To implement AES Encryption in Java, these are the packages that have been imported.

```
import java.security.SecureRandom;
import java.util.Base64;
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;
```

After importing these packages we proceeded to get the input from the user using the Scanner Class present in the Java Library. Now input text will be encrypted into a secret cipher text using javax.crypto.KeyGenerator class where the logic has been implemented to encrypt the given input (text form) into encrypted text in the form of string.

Post that, type casting the encrypted text into string using two string method and then capture the time-taken to encrypt the input text (in milli.secs) along with the number of characters in encrypted text.

B. Rivest-Shamir-Adleman (RSA)

To implement RSA Encryption in Java, these are the packages that have been imported.

```
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.util.Base64;
import javax.crypto.Cipher
```

After importing these packages we proceeded to get the input from the user using the Scanner Class present in the Java Library. Now input text will be encrypted into a cipher text using javax.crypto.KeyGenerator class where the logic has been implemented to encrypt the given input (text form) into encrypted text in the form of string.

Post that, type casting the encrypted text into string using two string method and then capture the time-taken to encrypt the input text (in milli.secs) along with the number of characters in encrypted text

C. Message-Digest 5 (MD5)

To implement MD5 Encryption in Java, these are the packages that have been imported.

```
import java.io.FileInputStream;
import java.io.UnsupportedEncodingException;
import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
```

After importing these packages we proceeded to get the input from the user using the Scanner Class present in Java Lib. Then A hash text has been generated with padding ‘1’ , based on the same hash text an user defined

function has been written where the logic has been implemented to encrypt the given input string. After a byte hash text is generated which is the hashed text for the given input and it's converted to String Format.

D. Symmetric hybrid Crypto System (AES + MD5)

To implement Symmetric hybrid Crypto System (AES + MD5) Encryption in Java, these are the packages that have been imported.

```
import java.security.SecureRandom;
import java.util.Base64;
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;
import java.io.FileInputStream;
import java.io.UnsupportedEncodingException;
import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
```

In this approach a single input will undergo two encryption levels. Firstly input will undergo AES encryption and gives an encrypted text which has the length corresponding to the input string length. Then this encrypted text has been converted into string and considered as the input for the MD5 user defined function to convert the AES generated encrypted text into hash text one more time with the given padding value to obtain final encryption message.

E. Asymmetric hybrid Crypto System (RSA + MD5)

To implement Symmetric hybrid Crypto System (RSA + MD5) Encryption in Java, these are the packages that have been imported.

```
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.util.Base64;
import javax.crypto.Cipher;
import java.io.FileInputStream;
import java.io.UnsupportedEncodingException;
import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
```

In this approach a single input will undergo two encryption levels. Firstly input will undergo RSA encryption and gives an encrypted text corresponding to the input string length. Then this encrypted text has been converted into string and considered as the input for the MD5 user defined function to convert the RSA generated encrypted text into hash text one more time with the given padding value to obtain final encryption message.

F. Multi-Level hybrid Crypto System (AES + RSA + MD5)

To implement multi-level hybrid Crypto System (AES + RSA + MD5) Encryption in Java, these are the packages that have been imported.

```
import java.security.SecureRandom;
import java.util.Base64;
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.util.Base64;
import javax.crypto.Cipher;
import java.io.FileInputStream;
import java.io.UnsupportedEncodingException;
import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
```

In this approach a single input will undergo three encryption levels. Firstly input will undergo AES and produces encrypted key of length corresponding to input string length and then AES encrypted key undergoes RSA encryption and gives an encrypted text corresponding to the input string length. Finally this encrypted text has been converted into string and considered as the input for the MD5 user defined function to convert the RSA generated encrypted text into hash text which makes the encryption message more robust to break.

V. CONCLUSION

The proposed Crypto System has been implemented in JAVA because of its existing application robustness. Also JAVA has built-in packages which are efficient and appropriate to implement the proposed system. The implementation approach has been confined to perform the encryption to meet the research-objectives. However, further research can be carried to perform the decryption as well in the future.

ACKNOWLEDGEMENT

I would like to thank Dr. Raghunatha Reddy for his invaluable support and guidance as a research guide to help me prepare and publish this paper. I would also like to extend my heartfelt thanks to Mr. E. Mahendranath Reddy (B.Tech. pursuant, REVA University, Bangalore) who helped me with implementation and testing of the proposed solution. Last but not the least, I would like to thank my spouse Parimala Devi and super Kids Dharani and Niyati for their understanding and unconditional support to complete this paper on time.

REFERENCES

[1] Data Security Optimization Using Hybrid Cryptography – http://ijsrd.com/Article.php?manuscript=IJSRDV7I120315
[2] Performance Assessment and Hypothesis Validation of Symmetric, Asymmetric & Multi-Level Hybrid Crypto Systems - http://ijsrd.com/Article.php?manuscript=IJSRDV8I20986

[3] Java se8 API cryptography documentation http://ijsrd.com/Article.php?manuscript=IJSRDV7I120315

