# Hybrid Encryption System

Sushant Susarla , Gautam Borkar

*Department of Computer Engineering MCT's Rajiv Gandhi Institute of Technology Andheri(W), Mumbai-53, India.*

***Abstract*— Encryption algorithms have been used since a long time to keep secret data safe from intruders. These algorithms are generally improved based on drawbacks of earlier algorithms. But each encryption algorithm has a drawback that is exploited very well by the intruders. These algorithms are so well published that their mechanism gets known to everyone and thus the loopholes are easily exploited. In order to overcome these drawbacks and in a bid to take out the best aspects from the famous algorithms and covering their flaws, in this survey, a technique of using multiple algorithms in a predefined order on the same set of data is suggested. That is, the algorithm security is greatly improved, through researching several famous data encryption algorithms, and improving some data encryption algorithms, and arranging encryption algorithms in some order.**

***Keywords*— AES, MD5, Encryption, Cypher, Hybrid encryption.**

## I. INTRODUCTION

With advancements in computer technology and the widespread reach of the world-wide-web i.e. the Internet, people's lives are changing rapidly. The liberalization, internationalization and personalization features of Internet have been attempting to bring revolutionary reform to government agencies, enterprises and institutions, at the same time help to boost the work efficiency and market response to improve their competitiveness by the use of Internet. But how to make the information system confidential and make sure that it is not leaked, even if they are stolen it is difficult to be identified, if they are identified after all the difficulty, they are extremely difficult to be modified. To prevent confidential information from being accessed, modified, fabricated, to keep that protected has become a hot research topic in the IT industry.

The encryption technology is the basic safety techniques used in current e-commerce and banking websites which are of extreme importance. Information encryption technology can not only meet the security requirements of confidentiality of information, but also avoid the leakage of the important information which are of high security especially in the security (defence) and hospital, banking sectors. Therefore, encryption technology is the base of authentication technology, as well as many other security technologies that are used today. At present, in different areas of software systems, the most popular encryption algorithms are the AES encryption algorithm, the MD5 encryption algorithm, the SHA1 algorithm, etc. In connection with characteristics of different encryption algorithms, we define a simple initial encryption algorithm first in this survey, then use the AES encryption algorithm, finally, form a hybrid encryption algorithm together with MD5 encryption algorithm. The hybrid encryption algorithm has greatly improved the security of the encryption algorithm.

## II. EXISTING SYSTEM

In the current encryption systems, individual algorithms are used to secure data. Such as Linux systems use MD5 encryption algorithm while some others use maybe AES or DES algorithms to encrypt their passwords. But each of these mentioned algorithms have been cracked some or the other time, which means they are not invincible and can be broken by a skilled hand. Thus the security of the data (passwords in many cases) is highly and threateningly compromised. All these algorithms are very famous all around the globe and are used by many, some are even open source. This means that the algorithm's flaws are well known to all and in some cases, even the source code is well known to many. This adds up to the security woes of these algorithms. Thus there needs to be a system which overcomes these drawbacks while upholding the positive aspects of these widely known algorithms.

## III. PROPOSED SYSTEM

The system proposed here intends to describe a hybrid system where encryption algorithms are used in a predefined order on the same set of data one after the other to finally obtain an encrypted data form. This encrypted data or cypher text can be used to transfer such confidential data without the fear of being rigged. The only way to decrypt such data would be to use the exact reverse order of the encryption process used in the encryption stage.

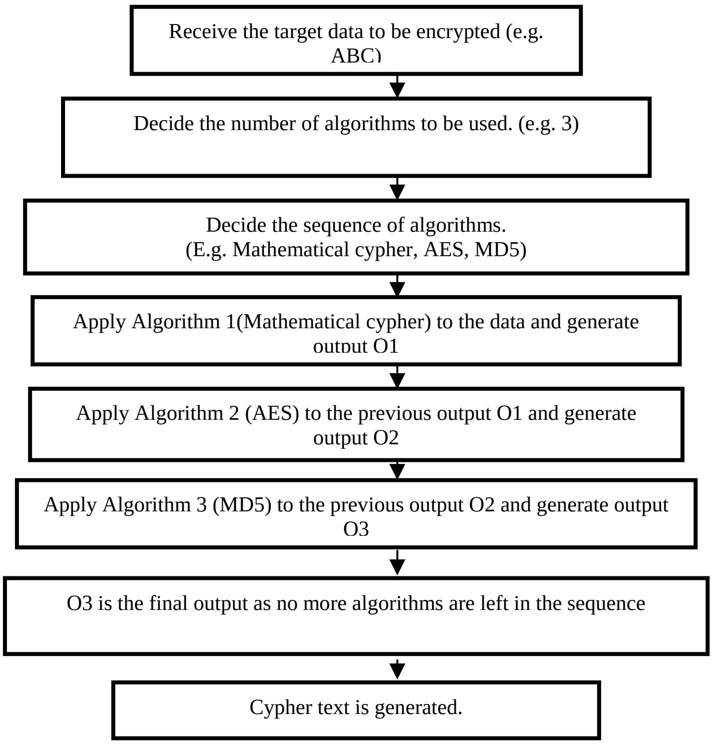
Receive the target data to be encrypted (e.g. ABC)
Decide the number of algorithms to be used. (e.g. 3)
Decide the sequence of algorithms. (E.g. Mathematical cypher, AES, MD5)
Apply Algorithm 1(Mathematical cypher) to the data and generate output O1
Apply Algorithm 2 (AES) to the previous output O1 and generate output O2
Apply Algorithm 3 (MD5) to the previous output O2 and generate output O3
O3 is the final output as no more algorithms are left in the sequence
Cypher text is generated.

Fig 1: Cypher text generation flow chart

The exact reverse application of the sequence of the algorithms needs to be used to decrypt the cypher text, thus even if the algorithms used in this process are widely known, unless the sequence is known to the decrypting body, decryption is impossible.

Example process with explanation:

Algorithms to be used are Mathematical cypher, AES and finally MD5.

Mathematical cypher:

Cypher algorithms were one of the most preluding algorithms of all the cypher algorithms. Most of them such as Caesar cypher, substitution cypher are inexistent today as they have fatal flaws or are too easy to decrypt. But the only possible way to crack these algorithms is if their output is known, because only then can the pattern be detected. But in hybrid system, the output of cypher algorithms are further encrypted by other algorithms and thus detection of pattern is impossible even if the attacker has the final cypher text. Mathematical cypher maybe or may not be used. But this may be used as a slight deterrent which would be producing output closest to the text and thus may act as a last step to lead the attacker haywire.

Let us consider the use of a mathematical cypher where each alphabet is substituted by its mirror alphabet (i.e. A changes to Z, B changes to Y and so on) if it is in odd place value else the digit remains the same.

Thus example data: ABC becomes ZBY.

AES encryption algorithm:

The Advanced Encryption Standard (AES), which implements the Rijndael cipher, is a symmetric block cipher that was developed as a result of a call by the United States National Institute of Standards and Technology in 1997 for a secure cryptosystem to replace the then standard Data Encryption Standard algorithm, which had become vulnerable to brute-force attacks.

AES is a symmetric algorithm. Therefore, it uses the same key for encryption and decryption. To allow for this phenomenon, all operations used in the encryption process must possess an inverse operation to exactly undo the transformations applied to the plaintext in order to recover the plaintext message from the cipher text at the time of decryption. A field obviously contains the additive and the multiplicative inverse of each of its elements and guarantees the existence of the inverse operation of any transformation (within the scope of operations available in the field) applied to its elements. Thus, a field satisfies the requirements that will allow use of the same key in the encryption and the decryption processes.

Theoretically, there is not a known way to crack the AES cryptosystem other than the use of brute-force attacks, but the actually implemented algorithm uses parts of the hardware that leave the system open to attacks. A side-channel attack could happen because each part of the implemented algorithm has its own runtime, so in theory, if the attacker can correlate runtimes with knowledge of the implementation, he could extract information about the key. It is only fair to say that these attacks are very difficult to attempt and practically impossible in real life, as there are many circumstances and knowledge of implementation details is required. Also, there are practices that can be used to defend against these attacks, like manipulating the kernel behaviour and trying to send information along specific CPU pipelines, but as stated earlier, the attack, by itself, is only theoretical and very unlikely to occur in the real world. Another planned attack to AES algorithm was the square attack, which was successful in breaking Rijndael's predecessor, a block cipher called Square. The square attack exploits the byte-oriented structure of the algorithm to extract information about the cipher key. This attack has been proven successful if the AES algorithm would have no more than seven rounds for the 128-bit key and no more than nine rounds for the 192- and 256-bit keys. However, with the current number of rounds for each possible key length, the square attack does not seem to threaten the security of AES unless we are able to reach the level of power necessary to break Rijndael cipher.

However, even with a few devices vulnerable to attacks, AES still remains one of the most secure cryptosystems, especially in applications with a larger memory space, which allows use of longer keys and additional masking in the key generation part of the algorithm.

Thus AES in itself is vulnerable to brute force and other forms of attacks and still is one of the most secure and used algorithms and thus if set into a hybrid algorithm set like the one, the brute force attacks on AES can be avoided as the output of AES would be further encrypted by other encryption algorithms and thus as the attacker wouldn't have the output of this algorithm in raw form, brute force attacks and other attacks that use output as a key to attack can be thwarted .

The output obtained in the previous algorithm acts as the input to AES i.e. ZBY acts as input to AES and the output obtained be: Èí]ÁbÓ½Òó)[?Û or (c8 ed 5d c1 1d 62 d3 bd d2 f3 29 18 16 5b 3f db) in hexadecimal form.[2]

This output is then given to the next algorithm to encrypt.

As we see that the output length varies from the input length, this disallows another vulnerability that may have resulted if the size would have remained the same.

MD5 encryption algorithm:

The MD5 message-digest algorithm is a widely used cryptographic hash function producing a 128-bit (16-byte) hash value, typically expressed in text format as a 32 digit hexadecimal number. It is used extensively by Linux systems. It compresses a piece of information with plain code and random length into 128 bits value by hash algorithm, which is called information distract. MD5 algorithm is irreversible and cannot recover the original plain code information from information abstraction, thus it is always believed safe. However, some researches indicate that MD5 algorithm could be deciphered by collision attack, and the security of its application has received the challenge.[4]

The main advantage of MD5 is that the output is a standard 128 bit irrespective of the input length and also that the output varies each time the data is encrypted even if the data is same, but the main disadvantage is that the output or cypher text may repeat after certain computations.

The output obtained in the AES algorithm is now used as input to MD5 algorithm and thus the input Èí]ÁbÓ½Òó)[?Û gets converted to 16d47741d5fe651278e114fa20b499cf which is the final output in this example simulation of the hybrid encryption algorithm technique.

## COMPARATIVE STUDY OF VARIOUS ENCRYPTION ALGORITHMS PLACED AGAINST HYBRID SYSTEM.

| Cypher Algorithm | Defects | Improvements |
|---|---|---|
| Transposition (Matrix) cypher | Transposition ciphers can be broken with restoring the original order of the letters. Column and row rearrangements, and frequency distributions of diagrams and trigrams (three-letter sequences) are commonly used in cryptanalysis of transposition ciphers | Transposition cyphers can be broken by various known rearrangement techniques, but the answers only make sense if the obtained texts after trials would make sense in any form. But if the trials despite being correct in decrypting the cypher text he wouldn't make sense, the purpose of the attacker would be defeated. Even if he is able to decrypt the cypher, he wouldn't understand that what he obtained is correct as the data he would obtain would be cypher text of another algorithm and thus the attacker wouldn't even know when he gets the correct text and when he doesn't get it. |
| Substitution cypher | An eavesdropper can decrypt a cipher- text by performing frequency analysis on the letters in the cipher text. Because of the substitution, the letter frequencies will be different than the frequencies for normal English text. The eavesdropper can still exploit this frequency information to break the cipher. | The frequencies match with the English alphabet frequencies which would be understood by the user, but in this case the breaking would result in an obscure text as the result obtained would be the output of another encryption algorithm. This would mean that even a simple substitution cypher, if used in a hybrid set, can prove dangerous as a simple 5 word substitution may yield $5^{26}$ possibilities, imagine a 128 bit cracking would be impossible even in a lifetime. |
| SHA1 algorithm | Analysis shows that collisions of SHA1 can be found with complexity less than 269 hash operations. This is the first attack on the full 80-step SHA1 with complexity less than the 280 theoretical bound. Based on estimation, it is expected that real collisions of SHA1 reduced to 70-steps can be found using today's supercomputers | The hybrid algorithm doesn't offer any significant improvement over the new SHA1 but the multiple security used due to multiple algorithms along with their random usage may provide extra security. |
| AES | AES is vulnerable to brute force attacks and also side channel attacks that focus on the run time of the algorithm to process the amount of steps used in encrypting the data. | Brute force is absolutely useless as brute force needs a confirmation that the answer is correct. In hybrid theory, the answers obtained too would be cypher texts of other algorithms and thus brute force wouldn't help at all. Side channel attacks would measure time only if one algorithm is used, but as we are using multiple algorithms and the attacker doesn't know their order as it is random for all the possibilities and thus time calculation is impossible. |
| DES algorithm | The main attacks possible on DES are automatic brute force and linear and differential attacks and this was the main reason that warranted the need of the use of AES over DES. | All these attacks on DES were possible due to its one time encryption but in the case of hybrid encryption, the data deciphered would too be a cypher text and thus brute force attack isn't possible. |
| DSS algorithm | Timing attacks have been the reason for the downfall of DSS in which the attacker takes a look at the time consumed and narrows down on the possible values of the secret key. | Timing attacks are only possible if there is only one algorithm being implemented, but in hybrid system, many algorithms are implemented that vary in the order of implementation and also the number of algorithms involved vary as per user and thus detecting time used itself isn't possible. |
| RSA algorithm | Boneh, Durfee and Frankel presented several attacks on RSA when an adversary knows a fraction of the secret key bits. The motivation for these so-called partial key exposure attacks mainly arises from the study of side-channel attacks on RSA. With side channel attacks an adversary gets either most significant or least significant bits of the secret key. | Detecting the MSBs and the LSBs of the code are only applicable if the derived deciphered text makes any sense to the attacker, but in case of hybrid algorithms, the deciphered text would make no sense to the attacker and thus he wouldn't be able to fathom which one is the real key though using this technique he would get near to decrypting the data as the MSBs and LSBs that he obtains would draw him closer to the conclusion but the multi-layer security would prove a great deterrent. |
| MD5 | MD5 algorithm is vulnerable to Collision and dictionary attack. The attacker would get the plain text one he successfully deciphers the MD5 | In hybrid algorithms, even though the dictionary attack and collision attack threats are not resolved and the attacker manages to crack open and decrypt, all he gets is another cypher text of the algorithm used just before MD5. This successfully adds another layer of security despite getting broken on one front. |
| Affine cryptosystem | There are three major possible attacks for the affine cipher:, they are<br>1. Frequency Attack: Since the frequencies of the various diagrams in English have been tabulated.<br>2. Exhaustive Search: Each plaintext letter has only five possible corresponding Cipher text letters.<br>3. Cipher text only attack: unless the keyword is long, the last few rows of matrix are predictable. | The affine crypt encrypts the data using a matrix and thus the only way of attack is frequency analysis or brute force. But in hybrid algorithms, the affine system's output would be encrypted by some other algorithm such as say MD5 and thus the characters wouldn't remain same in number as the letters in the input(128 bit standard output by MD5) and thus both the attacks get nullified and cannot be implemented. |

## IV. Future Scope

This system is flexible to be implemented by the user in many ways as he desires i.e. the number of algorithms to be used, the sequence of algorithms and even the algorithms to be used may vary from user to user. This ideally makes it strenuous for the attacker to attack or decipher the plain input text unless he knows the algorithms used in the process of encryption, the sequence they are used in, etc. as the sequence is vital for decryption. This flexibility enables a wide range of uses and also enables accommodation of new algorithms as and when they are developed in the future.

In another innovative approach, a password may be set which would help decide the algorithms to be used from an array of them and also the sequence of those algorithms to be used would be decided by the unique password which would also be needed for decryption. A software can be built for this purpose consisting many algorithms and then as soon as it receives password and data, it computes the algorithms to be used and their sequence based on the password and apply it to the data and during decryption, follow the same process and input the encrypted text and password to decrypt. This would improve the efficiency of the hybrid system greatly.

E.g. let the software have 9 algorithms in it (such as AES, DES, MD5, etc.).

Let the password be 1542

This means that the algorithms 1, 2, 4 and 5 will be used in the sequence 1-5-4-2 to implement the hybrid encryption system.

Thus until the password is known, the decryption will be impossible and the complexity would increase with the increase in the number of algorithms used.

## V. Conclusions

Using a single encryption algorithm renders it vulnerable to many attacks that attack the weak chinks in the algorithm's armour especially most algorithms face grave danger against brute force attacks. Thus using multiple algorithms in a sequence where the output of one algorithm is the input for the next algorithm in the series provides extra security by securing the data exponentially well and the flexibility of the technique as mentioned before makes it wonderful to be used with future algorithms and also to be implemented with passwords for extra protection. The multiple encryptions along with the randomness in nature of the selection of the algorithms, their sequence and the number of algorithms used provides a lot of safety in multiple layered manner as compared to the algorithms used as a stand-alone entities and guards against many well-known attacks. Classical algorithms that were easily cracked but had a easy way to encipher can be used in very efficient way if included into hybrid algorithms. The main and only negative is the overhead and the time and resources used for multiple encryptions would be huge and would keep increasing with the number of algorithms used.

## References

[1] 2012 Fourth International Conference on Computational Intelligence and Communication Networks. - The Application of Hybrid Encryption Algorithm in Software Security.

[2] AES encryption - http://people.eku.edu/styere/Encrypt/JS-AES.html.

[3] MD5 hash generator- http://www.md5hashgenerator.com/index.php

[4] Research for the Application and Safety of MD5 Algorithm in Password Authentication- 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery

[5] IEEE publications (digital library).

[6] Research for the Application and Safety of MD5 Algorithm in Password Authentication-9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2012) .

[7] A. J. Menezes, P. C. Van Oorschot, and S. A. Van Stone, Handbook of Applied Cryptography, CRC Press, 1997.

[8] New Partial Key Exposure Attacks on RSA- Faculty of Computer Science, Electrical Engineering and Mathematics Paderborn University 33102 Paderborn, Germany.

[1] Timing Attacks on Implement at ions of RSA, DSS, and Other Systems- Paul C:. Kocher Cryptography consultant(, P.O. Box 8243, Stanford, CA 94309, USA.