New Approach and Additional Security to Existing Cryptography Using Cubical Combinatorics

P.Elayaraja[1], M.Sivakumar[2]
Master of Computer Applications
Dhanalakshmi College of Engineering, Chennai
pelayaraja1@gmail.com ,msivakumara@gmail.com

ABSTRACT

This paper proposes a new approach to information security using cryptography which generates keys on cubical combinations. Peoples use cubes for multiple purposes. Rubik's Cube is a cube shaped puzzle. Rubik's cube has been used as a play think without any significant applications. But this cube can be used to develop a new field of cryptography called, Cubical Combinatorial Cryptography.

This paper deals with key Generation and Cryptography using the mathematical concept of combinatorics provided over a NXNXN Rubik's cube shuffled to random position. The key generated is of a specific structure and is practically impregnable and immutable. This concept can be used for the secure transfer of keys, encryption and decryption of data. This paper also verifies the strength of the keys generated using the combinatorics method.

The fact that this mode of key generation has never been explored or broken into, makes this approach new, and yet another effective method. The advantages of this new algorithm over the others are also explained.

KEYWORDS:
Security, Rubik's cube, Symmetric key cryptography, Combinatorics

1 INTRODUCTION

People have always bean fascinated with the idea of hiding information from others .Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables you to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot be read by anyone except the intended recipient. It is used to Data privacy and Data integrity and authenticity such as protect our financial information from thieves, to protect our personal information from marketing companies, and in some cases it's even used to protect individuals' freedoms from malicious governments. Cryptography is, indisputably, one off the most important fields within the security profession. Unfortunately, it also seems to be the least understood.

MODERN CRYPTOGRAPHY:

Modern Cryptography abandons the assumption that the Adversary has available infinite computing resources, and assumes instead that the adversary's computation is resource bounded in some reasonable way. In Particular, it may be assumed that the adversary is a probabilistic algorithm which runs in polynomial time. Similarly, the encryption and decryption algorithms designed are probabilistic and run in polynomial time. The running time of the encryption, decryption, and the adversary algorithms are all measured as a function of a security parameter k which is a parameter which is fixed at the time the cryptosystem is setup.

ALGORITHM AND KEY DESIGN ISSUES

If the adversary algorithm[1] runs in polynomial time ,it means time bounded by some polynomial function in k .According ,in modern cryptography, the infeasibility of breaking the encryption system and computing information about exchanged messages are dealt, where as historically, the impossibility of braking the encryption system and finding information about exchanged messages is dealt. It is noted that the encryption system which is decrypted and claimed "secure" with respect to the new adversary are not "secure" with respect to a computationally unbounded adversary in the way that the one-time pad system was secure against an unbounded adversary. But ,on the other hand, it is no longer necessarily true that the size of the secret key that Alice and Bob meet and agree and before remote transmission must be as long as the total; number of secret bits ever to be exchanged securely remotely. In fact, at the time of initial meeting, Alice and Bob do not need to know in advance how many secret bits

they intend to send in the future. This paper shows how to construct such encryption systems, for which the number of messages to be exchanged securely can be a polynomial in the length of the common secret key.

CRYPTOGRAPHIC PRIMITIVES

As modern cryptography is based on a gap between efficient algorithms for encryption for the legitimate user versus the computational infeasibility of decryption for the adversary, it requires that available primitives with contain special kind of computational hardness properties are employed. Of these, perhaps the most basic is a one–way function. Informally, a function is one-way if it is easy to compute but hard to invert. Other primitives include pseudo random number generators, and pseudo random function families .From such primitives, it is possible to build secure encryption schemes. Thus, a central issue is where these primitives come from. Although one-way functions are widely believed to exist, and there are several conjectured candidate one-way functions which are widely used, it is not known how to mathematically prove that they actually exist. Hence this paper suggests a design of a cryptographic scheme assuming that a one-way function is given.

2 THE PRESENT CRYPTOSYSTEMS

SECRET KEY CRYPTOGRAPHY

Secret key cryptography, a single key is used for both encryption and decryption. The sender uses the key (or some set of rules) to encrypt the plaintext and sends the ciphertext to the receiver. The receiver applies the same key (or rule set) to decrypt the message and recover the plaintext. Because a single key is used for both functions, secret key cryptography is also called *symmetric encryption.* Secret key cryptography schemes are generally categorized as being either *stream ciphers* or *block ciphers.*

STREAM CIPHERS

Stream ciphers are usually the combination of pseudo-random key information with plaintext for encryption, or cipher text for decryption one bit at a time. This is usually done using the XOR operation. Because stream ciphers are typically much faster than other types of ciphers, they are used when encrypting phone calls or network traffic. Stream ciphers are symmetric key algorithms, meaning that the encryption key and decryption key are the same. Common stream ciphers include Rc4 and A5/1.

BLOCK CIPHER

Block cipher typically take a block of input, perform an operation (encryption or decryption) and output a same-sized block. Thus, all block ciphers have a natural block size - the number of bits they encrypt in a single operation [9]. For example, when encrypting, a block cipher will read a block of plain text operate on it using the key and output a block of cipher text. When decrypting, it reads a block of cipher text, operations on it using the key, and outputs a block of plain text. Additionally, block ciphers may be run in a variety of modes which affect the operations. Some block ciphers may even be used as stream ciphers, however they are typically slower than actual stream ciphers. Common block ciphers include AES, DES, and IDEA.

PUBLIC KEY CRYPTOGRAPHY

Public key cryptosystem [2] are unique, in that they use different keys for encryption and decryption. This is based on a relationship between three numbers: the encryption key and decryption key, and the modulus. Encryption operates on plaintext using the encryption key and the modulus to produce cipher text. Decryption operates on cipher text using the decryption key and the modulus to produce plaintext. One's public (encryption) key may be widely distributed without fear of compromising messages encrypted with it. As long as the private (decryption) key is a kept secret, the communications are secure.

Asymmetric cryptography solves several problems inherent in symmetric cryptography, such as key exchange over insecure channels, authentication, and non-repudiation using digital signatures. Public key encryption is slow. For this reason, most implementation of asymmetric encryption use the technology to encrypt a randomly generated session key that is then used to encrypt and decrypt the plain text with either a block or stream cipher. Common public key cryptosystem include RSA, Digital Signature Algorithm (DSA), Cramer-Shoup, Diffie-Hellman, and EIGamal [10].

ELLIPTIC CURVE CRYPTOGRAPHY

Elliptic curve cryptography is the next generation of public key cryptography that uses elliptic curves to reduce numbers instead of modules.

One effect of using elliptic curve cryptography is that the key length required is much closer to those for block and stream chippers. At this point in time, elliptic curve cryptography is relatively new, and has not been sufficiently analyzed for practical use.

QUANTUM CRYPTOGRAPHY

The term quantum cryptography actually refers to a key exchange method, not an actual cryptosystem or type of cryptosystem. Quantum cryptography relies on the laws of physics to ensure that eavesdroppers are unable to successfully gain access to the key while it is in transit. While this type of key exchange protocol is very promising, it is not very practical for widespread use at this time.

KEY LENGTHS

Key length is directly proportional to the security of the cryptosystem. However, like all things in security, key length is a tradeoff. Each additional bit of a key exponentially increases the length of time required to perform a brute force attack against it. On the other hand, each bit also adds to the time required for encryption and decryption. It is for this reason that everyone isn't using keys that are millions of bits in length. Further home, different t types of cryptosystems require different key lengths FOR SIMILAR LEVELS OF SECURITY. Most public key cryptosystems require vastly longer keys than block or stream chippers. For instance, the current recommended key length of RSA is 2048 bits, while the current recommended key length for block or stream chippers is 128 or 256 bits. This discrepancy is based on the different types of problems that need to be overcome to break the encryption .Unlike block or stream chippers, public key cryptosystems generally upon the difficulty of factoring large numbers or determining discrete algorithms. While these problems are still considered highly difficult, key length should increase as advancements in both the problems themselves and computing power come about.

DIGITAL SIGNATURES

The advent of public key cryptography brought about great changes in the security world. Suddenly problems that had appeared to have no solution, such as non-repudiation, were easily overcome .Just as anyone could encrypt a message with someone else's private key, they could easily encrypt it with their own. A normal use of public key encryption looks like this .Alice encrypts a message with Bob's public key. A digital signature [4] is the same operation with different keys. Alice encrypts a message with her own private key. Bob decrypts a message with Alice's public key.

As long as Alice's private key has not been compromised, the message can be validated as having come from Alice. Another, more common, way to accomplish this is to take a cryptographic hash of the message, and encrypt it with the sender's private key.

ONE WAY HASH FUNCTION

Hash functions, also called *message digests* and *one-way encryption*, are algorithms that, in some sense, use no key [10].One-way hash functions or cryptographic hashes are often used in digital signatures, and have the following attributes:

- No two messages produce the same hash
- It is infeasible to derive the original message from a hash
- It is infeasible to produce a message that hashes to a given value

In this case, Alice hashes a message, and encrypts the hash with her private key. This signature is then appended to the messages. Hashes, like keys, are measured in bits. Common cryptographic hash functions include MD5, RIPEMD, HAVAL (HAsh of VAriable Length), Whirlpool, Tiger, SHA-1, and SHA-256.Due to problems discovered in MD5 and SHA-1 stronger hash functions can be used with larger hashes, such as SHA-256.

3 PROPOSED THEROY

INTRODUCTION

The key encryption is symmetric in nature. The key generation and encryption is based on the cubical values of the well known Rubik's cube. The basic attributes of the key generation are

1. Arrangement of letters on the faces
2. arrangement of sides
3. Arrangement of colors
4. steps to reach a solved cube
5. Cube degree

FORMING THE KEY

Select the cube degree based on the data size. If the size is DATA_SIZE select in such that, n is minimum when the equation, **6n^2>DATA_SIZE** is solved. If size is DATA_SIZE=500 bits select n=10. So the objective is to design a 10 x 10 x 10 cube. Select Face order (FO). The face are named as front

(1), up (2), right (3), down (4), left (5) and back (6). So the no of faces order combinations is 6!.The no of bits to represent this is 16 bits. The colors are read in the arrangement order (AO). There are different AO's possible. Some are North West Circular AO (NWCAO), North West Vertical AO (NWV-AO), North West Horizontal AO (NWH-AO), Center Right Circular AO (CRC- AO) and Center Left Circular AO (CLC-AO). The color AO also is used for data coding and or reading there are totally 6 colors on the cube. Giving each color a numeric value (Called Color Values, CV) such as, green (1), Pink (2), Yellow (3), Blue(4), Violet (5), and Orange (6).The Shared_Secure_Key(SSK) is designed by adding the critical_data which given in the encryption process. The number of colors exposed in the key is inversely proportional to the security parameter of the algorithm. More than number of colors exposed higher is the chance of designing the initial state of the cube, thereby decreasing the intended security.

Assuming the exposed colors are Green and Blue, code by the AO and FO as 104001000040404000110000(For a 2x2x2cubein NWH-AO and 123456-F0).The SSK is manually transferred by some secure means. The unexposed colors are filled with a '0'.

ENCRYPTION PROCESS

Now the data is fed into cube faces or the Feasible Data Fields. The following figure1 shows the data "BEST WISHES!" in NWH-AO and 231654-FO.

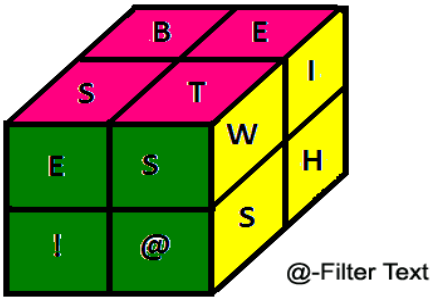


Fig 1.

The main process of encryption starts here. The scrambled cube with the data is solved by any globally known or unknown algorithm and the steps are noted. This sequence of steps is called the Critical Data. The layered approach algorithm defines the FO operations as, Clock-wise front (1) and Anti-clockwise front (1) or front inverse ('1'or -1).Similarly for all FOs the clockwise is it value and the anticlockwise is negation of its value. A critical data can be of this form:

[6'|1|5'|3|4'|3'|4|1|6|2|6'|2'|1'|4|5]

This means that taking these steps leads to the solving of the cube as in the fig2.

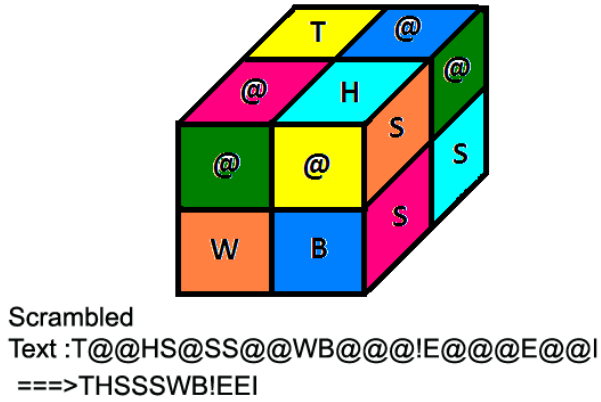


Fig 2.

DATA TRANSFER

A secure way to sending the critical_data is designed. A message digest is created for the critical_data using the SHA-1 algorithm. Here the SHA was the existing system.

The critical data,[6'|1|5'|3|4'|3'|4|1|6|2|6'|2'|1'|4|5] can be written in hexadecimal format as e1d3cb4162ea945 and the SHA-1 hash generated is 70f5db2d4d3f3d7b42d07f0dacccce7f97ddeb6a.

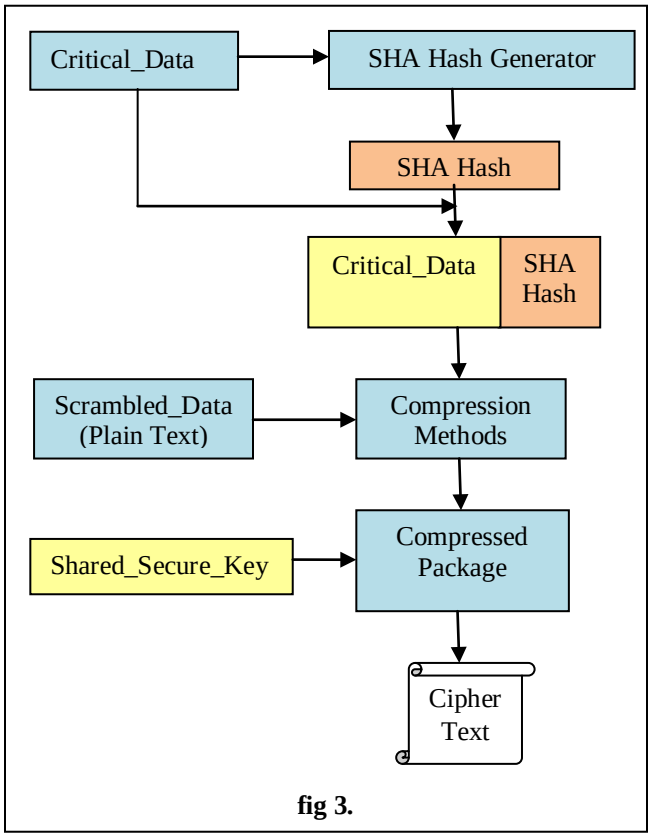
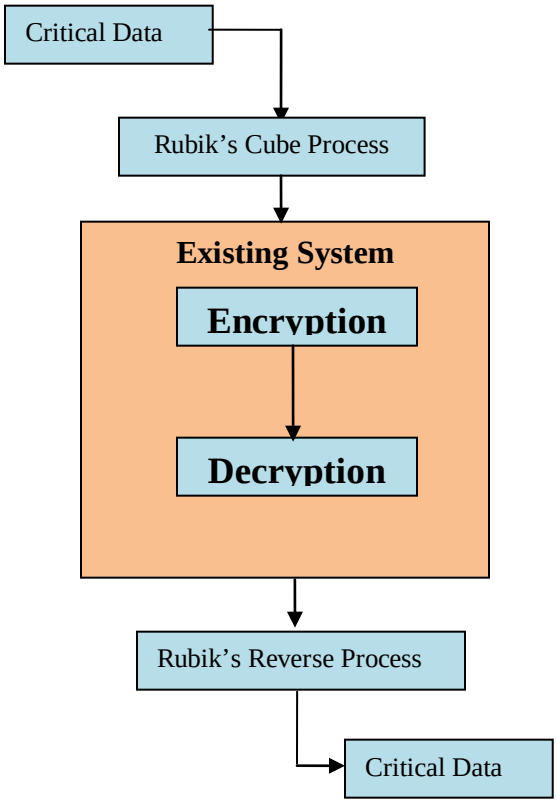


fig 3.

The encrypted data is transferred through normal means. In order to provide extra level of security to the encrypted data, the procedure followed for the critical_data can be employed here.

PROPOSED ARCHITECTURE



DECRYPTION PROCESS

The decryption process starts here. The client is provided with the following Shared_Secure_Key, the 'pocket', the encrypted data. He is also aware of the following: Cube degree, Face Order (FO), Arrangement Order (AO), Color Values (CV). The decryption process is as follows. The encrypted_data is filled in the cube using the FO and FO used. If the extra layer of security similar to that of the critical_data was provided, then, the decryption process has to follow the extra steps to retrieve the encrypted_data. The packet is decrypted using the Shared_Secure_Key, the SHA-1 has is removed (and decompression is done before, if packet is compressed), and the critical_data is got.

Using the 'reverse (Critical_data)' function, (which does the opposite moves to solve the cube to its original position where it started) the cube is operated and original position reached. Using the Shared_Secure_Key which is the CV ordering, the authentication of the cube pattern can be checked. Finally using the FO and AO, the data is read.

The attackers cannot recover the original data until use the proper decryption process.

4 COMBINATION COMPLEXITIES

PERMUTATIONS

A normal (3x3x3) Rubik's cube has eight corners and twelve edges. There are 8! Ways to arrange the corner cubes. Seven can be oriented independently, and the orientation of the eight depends on the preceding seven, given 3^7 possibilities. There are 12!/2 ways to arrange edges, since an odd permutation of the corners implies an odd permutation of the edges as well. Eleven edges can be flipped independently, with the flip of the twelfth depending on the preceding once, giving 2^{11}

$8! \times 3^7 \times 12! \times 2^{10} \approx 4.33 \times 10^{10}$ possibilities.

There are exactly 43,252,003,274,489,856,000 possibilities [8], which is approximately forty-three quintillion (short scale) or forty-three trillion (long scale).The puzzle is often advertised as having only "billions" of positions, as the larger number could be regarded as incomprehensible to many. To put this intro perspective, if every permutation of a 57-millimeter Rubik's Cube were lined up end to end, it would stretch out approximately 261 light years.

Despite the vast number of positions, all Cubes can be solved in twenty-five or fewer moves. This fact helps us to encrypt easily. The large number of permutations is often given as a measure of the Rubik's cube's complexity. However, the puzzles difficulty does not necessary follow from the large number of permutations .The problem of putting the 256 letters of the alphabetical is alphabetical order has a larger complexity ($26!=4.03 \times 10^{26}$ possible orderings), but is less difficult.

BASE CASE COMPLEXITY

Assuming 25 moves (maximum moves taken to solve a cube) takes 10.23 seconds (minimum time ever recorded to solve a cube) manually, 1 move is computed in 0.41 seconds.

Let us assume 1 move is computed in 1 nanosecond by the computer. So a maximum of 4.33*10^19 possible positions takes an average time of 137 years. So encrypting the message will be easy where as decrypting (where we need to get the unscrambled cube) will take a long time without the Critical_data.

5. ACHIEVING SECURITY:

The Shared_Secure_Key provides security by reducing the number of critical parameters (Which are to be shared) that decide the ciphering properties. It also serves the purpose of verification of the initial cube faces. Moreover confidentiality is also achieved as this the only way to retrieve the hashed critical_data.

Hashing mechanism provided to the critical_data ensures integrity, non-repudiation and also confidentially. This mechanism when followed for the encrypted_data helps in enhancing the security of the data to be sent.

Still higher levels of security can be attained by making the AO and FO as private between the sender and receiver.

6 CONCLUSION & FUTURE WORK

In this paper a new concept combining cryptography and Rubik's cube has been introduced for information security and network security. Detailed steps with illustrations, of the concept have been described. Also the strength of the algorithm is discussed by explaining the complexities in encryption and decryption. This concept can be further enhanced by adding digital signatures to the cipher and the key transfer process. Also to avoid the evaluation complexity in the higher order cubes, the data can be fragmented and multiple lower order cubes can be processed in a distributed manner. This concept can be used for basic applications such as credit card transactions, pin transfers, bank account management, password management etc. Complex applications like defense data transfer which require high level security rather than lower time complexity can be dealt using this method operated over higher degree cubes.

REFERENCES

[1] Shafi Goldwasser and Mihir Bellare, " Lecture Notes on c", MIT Laboratory of computer Science, University of California at San Diego August 2001

[2] Garfinkel, S.L. "Public Key Cryptography" Computer Volume 29, Issue 6, June

[3] Toytan, M. "Quantum Cryptography" Signal Processing and Communications applications, 2007.IEEE 15th 11-13 June 2007 Page(s):1-4

[4]Lee, W.-B., Chang, C.-C. And Harn, L."Comment on Digital signature with (t,nI) shared verification based on discrete logarithms [and reply]" Electronics Letters Volume 31, Issue 3, 2 Feb. 1995 Page(s):176-177

[5]Chuan Zhou, gemei Zhu , Baohua Zhao and Wei "Study of one-way Hash Function to Digital Signature Technology" Computational Intelligence and Security, 2006 International Conference on, Volume 2, 3-6 Nov 2006

[6] Donald E. Eastlake, and Paul E. Jones "RFC3174–US Secure Hash Algorithm 1 (SHAI)"

[7] "Cryptography and Network Security 2nd Edition" Atul Kahate

[8] http://en.wikipedia.org/w/index.php?title=Rubik%27s_cube&redirect=no

[9]www.freesoft.org/CIE/Topics/143.htm

[10]www.garykessler.net/library/crypto.html