IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

An Imperative repetition for Disaster Management Communication System

Shipra Srivastava
Research Scholar
Dr. K N Modi University, Newai
Rajasthan, INDIA
shiprasrivastava2000@gmail.com

Dr.Ramveer Singh
Professor, Departmet of Information Technology
Greater Noida Institute of Technology,
Greater Noida, U.P., INDIA
ramveersinghrana@gmail.com

Dr. Anoop Tiwari
Professor, Department of C.S.E.
Dr. K N Modi University, Nawai
Rajasthan, INDIA

Abstract : **Disaster Management is too much tedious task for humans, as life challenging. Disaster like (Earthquake, Flooding, Volcano eruption, Land sliding or like Tsunami etc.) gives always a new life after finishing the previous complete life. In this article we approach a system for prevent disaster as well as recovery after Disaster. Today's era without communication, the life is almost not possible to continue in form of luxuries and satisfaction, we introduce a new approach for secure and error free communication for Disaster Management.**

Index Terms **:** Disaster, ORDES, Fuzzy Error Correction Code, Communication Security.

I. Introduction

DI-MAN-C (Disaster Management Communication) is a system that will help out all the persons in form of Disaster Management or Disaster Recovery. We introduce a system for Disaster Management using various current technology. Basically it is an architecture for implementation using communication network MANET, Mobile Agent and Sensors.

In this architecture the system follow all the basic aspects of communication system and also communicate with our helping hands like (Police Department, Fire Department and Disaster Recovery etc.). The flow of information is necessary to know the system Ex. Information for Recovery agencies as could as various departments for downwards and sending report to higher

authority as upward flow for making efficient decision to instruct the agencies and other Disaster Management. In this architecture we analyses the Disaster if the disaster is natural like (Earthquakes, Tsunami, Cyclone, Land

Slides) Or unnatural like (Terrorist Attacks, or any individual attempt) The system always gives the effective solution and provide help for person, thing resource which are located at Disaster Site. Basically the system is a WAN i.e. Wide Area Network using high effective sensors with connecting sub networks like Personal Area Network (PAN), Storage Area Network (SAN). Using SAN we create a server which will work on OLAP, OLTP. The server consists of all information for Disaster and have some authority to generate effective decision with help of Artificial Intelligence expert systems.

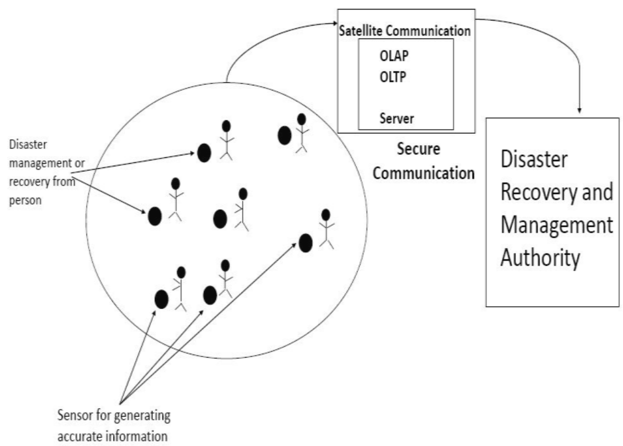

Fig 1. Architecture of communication network

The complete system are divided into three parts:

1. Disaster Site
2. Disaster Management Authority
3. Communication Channel (Secure and Error Less)

II. PRELIMINARIES

2.1 Disaster:

2.1.1 Natural disaster

A natural disaster is a major adverse event resulting from natural processes of the Earth. Disasters are:

1. Avalanches
2. Earthquakes
3. Volcanic eruptions
4. Hydrological disasters
    - 4.1. Floods
    - 4.2. Limnic eruptions
    - 4.3. Tsunami
5. Meteorological disasters
    - 5.1. Blizzards
    - 5.2. Cyclonic storms
        - 5.2.1. Tropical Cyclones
        - 5.2.2. Extratropical Cyclones
    - 5.3. Droughts
    - 5.4. Hailstorms
    - 5.5. Heat waves
    - 5.6. Tornadoes
6. Wildfires
7. Health disasters
    - 7.1. Epidemics
8. Space disasters
    - 8.1. Impact events
    - 8.2. Solar flare

2.1.2 Unnatural disaster

Unnatural disasters are the deaths and damages that result from human acts of omission and commission.

2.2 ORDES Approach:

We also using the same feistel structure and same process for encryption and decryption but we add a new process for key generation. In this process, key itself generate n different keys using a function and random number generated by Hardware Random Number Generator (HRNG) then new generated key block applies on the each block of message for all round of DES. For each block of message, the process generates a separate key. This new generated key used in encryption phase as well as the decryption phase.

M = {m1, m2, m3, ..., mn} (64- bit block of message)

K = {K} (56- bit Key)

2.2.1Key Generation:

F {K and Rj} = [K_new (x)_p]

Where Rj generated by Hardware Random Number Generator (HRNG) and [1 ≤ Rj ≤ (256 = 72,057,594,037,927,936)].

[K_new (x)_p)] = [K_new (x)_1] to [K_new (x)_n]

Function F:

Step 1- Input the bit value of initial key K (56-bit).

Step 2- Input generated random number Rj , generated by HRNG*.

(*HRNG Property- 256 no., random number generator)

Step 3- Convert Rj into 56- bit binary number.

Step 4- Now, we have

Key K = {KB1, KB2, KB3, ..., KB56}

And Rj ={Rb1, Rb2, Rb3, ..., Rb56}

Where KBr is the bit of Key and Rbr is the bit of Random no.

Here r =1, 2, 3..............56.

Step 5- Apply condition on K and Rj.

IF Rbr = 1 then, Complement (convert 1 to 1 or 0 to 0) of corresponding KBr.

ANDIF Rbr = 0 then, Retain the same (1 to 1 or 0 to 0) of corresponding KBr.

Step 6- [K_new (x)_p] = Result of step 5.

Using this function F every time we get the result [K_new (x)_p] for each block of message. For each block of M we generate a new no. Rj and implement function F. Finally get a new key for every block of message.

2.2.2 Encryption/Decryption:

In encryption phase, ORDES take a message block mn and a new generated key [K_new (x)_p], implement encryption process as per traditional DES. One special thing make aur process is different- REKEYING PROCESS.

REKEYING PROCESS have the property to make various key for various block of message.

Now, we have a new key for every block of message. This new key [K_new (x)_p] is apply on each block of message M. In this process, new key is also make 16 different key for every round of DES using shifting property as per traditional DES. For every block of message M, new key [K_new (x)_p] makes a new key block for every round of DES to implement in the encryption process.

Decryption Process is the inverse step of encryption process. In decryption, we also use the same key which is used in encryption.

Ci = E_([K_new (x)_p)] {mi} and mi = D_([K_new (x)_p]) {Ci}, where 1≤ i ≤ n.

Cipher Text C = {C1, C2, ..., Cn} and

Plain Text M = {m1, m2, ..., mn}.

2.3. Crisp Commitment Schemes

In a commitment scheme, one party sender aim to entrust a concealed message m to the second party receiver, intuitively a commitment scheme may be seen as the digital equivalent of a sealed envelope. If Sender wants to commit to some message m, sender just puts it into the sealed envelope, so that whenever sender wants to reveal the message to receiver, sender opens the envelope. First of all the digital envelope should hide the message from: receiver should be able to learn m from the commitment. Second, the digital envelope should be binding, meaning with this that sender can not change her mind about m , and by checking the opening of the commitment one can verify that the obtained value is actually the one sender had in mind originally.

2.3.1 Definition: A Commitment scheme is a tuple$\{P, E, M\}$ Where $M = \{0,1\}^n$ is a message space, P is a set of individuals , generally with three elements A as the committing party, B as the party to which Commitment is made and TC as the trusted party , $E = \{(t_i, e_i)\}$ are called the events occurring at times t_i , $i = 1,2,3$, as per algorithms e_i , $i = 1,2,3$. The scheme always culminates in either acceptance or rejection by A and B.

The environment is setup initially, according to the algorithm$setupalg\ (e_1)$ and published to the parties A and B at timet_1.

During the Commit phase, A uses algorithm$commitalg\ (e_2)$, which encapsulates a message$m \in M$, along with secret string $S \in_R \{0,1\}^k$ into a string$c$. The opening key (secret key) could be formed using both m and$S$. A sends the result c to B (at timet_2).

In the Open phase, A sends the procedure for revealing the hidden Commitment at time t_3 , and B uses this. $openalg\ (e_3)$: B constructs c' using $commitalg$, message m and opening key, and checks weather the result is same as the commitment c .

Decision making:

If $c = c'$.

Then A is bound to act as in m

Else he is free to not act as m

2.3.2 Definition: Fuzzy Commitment scheme is a tuple $\{P, E, M, f\}$ Where $M \subseteq \{0,1\}^k$ is a message space which consider as a code, P is a set of individuals , generally with three elements A as the committing party, B as the party to which Commitment is made and TC as the trusted party , f is error correction function (def. 2.2.5) and $E = \{(t_i, e_i)\}$ are called the events occurring at times t_i , $i = 1,2,3,$ as per algorithms e_i , $i = 1,2,3$. The scheme always culminates in either acceptance or rejection by A and B.

In the setup phase, the environment is setup initially and public commitment key CK generated, according to the algorithm$setupalg\ (e_1)$ and published to the parties A and B at timet_1.

During the Commit phase, Alice commits to a message $m \in M$ according to the algorithm $commitalg\ (e_2)$ into string$c$.

In the Open phase, A sends the procedure for revealing the hidden Commitment at time algorithm t_3 and B uses this algorithm $openalg\ (e_3)$: B constructs c' using algorithm$commitalg$, message t (m) and opening key, and checks weather the result is same as the received commitment$t(c)$, where t is the transmission function. Fuzzy decision making:

If $(nearest\,(t(c), f(c')) \le z_0$

Then A is bound to act as in m

Else he is free to not act as m

2.3.3 Definition: A metric space is a set C with a distance function $dist : C \times C \rightarrow R^+ = [0, \infty)$, which obeys the usual properties(symmetric, triangle inequalities, zero distance between equal points).

2.3.4 Definition: Let $C\{0,1\}^n$ be a code set which consists of a set of code words c_i of length n. The distance metric between any two code words c_i and c_j in C is defined by

$$dist(c_i, c_j) = \sum_{r=1}^{n} |c_{ir} - c_{jr}| \qquad c_i, c_j \in C$$

This is known as Hamming distance.

2.3.5 Definition: An error correction function f for a code C is defined as $f(c_i) = \{c_j \,/\, dist(c_i, c_j)$ is the minimum, over $C - \{c_i\}\}$. Here, $c_j = f\left(c_i\right)$ is called the nearest neighbor of c_i .

2.3.6 Definition: The measurement of nearness between two code words c and c' is defined by nearness $(c, c') = dist(c, c')/n$, it is obvious that $0 \le$ nearness $(c,c') \le 1$.

2.3.7 Definition: The fuzzy membership function for a codeword c' to be equal to a given c is defined as

$$FUZZ(c') = 0 \qquad \text{if nearness}(c, c') = z \le z_0 < 1$$
$$= z \qquad \text{otherwise}$$

2.4 SEQUITUR Algorithm

The SEQUITUR algorithm represents a finite sequence _ as a context free grammar whose language is the singleton set {σ}. It reads symbols one-by-one from the input sequence and restructures the rules of the grammar to maintain the following invariants:

(A) no pair of adjacent symbols appear more than once in the grammar, and

(B) every rule (except the rule defining the start symbol) is used more than once. To intuitively understand the algorithm, we briefly describe how it works on a sequence 123123. As usual, we use capital letters to denote non-terminal symbols. After reading the first four symbols of the sequence 123123, the grammar consists of the single production rule S ➔ 1, 2, 3, 1 where S is the start symbol. On reading the fifth symbol, it becomes S ➔ 1, 2, 3, 1, 2 Since the adjacent symbols 1, 2 appear twice in this rule (violating the first invariant), SEQUITUR introduces a non-terminal A to get

S ➔ A, 3,A A ➔1, 2

Note that here the rule defining non-terminal A is used twice. Finally, on reading the last symbol of the sequence 123123 the above grammar becomes

S ➔ A, 3, A, 3 A ➔ 1, 2

This grammar needs to be restructured since the symbols A, 3 appear twice. SEQUITUR introduces another non-terminal to solve the problem. We get the rules

S ➔ B,B
B ➔ A 3
A➔ 1 2

However, now the rule defining non-terminal A is used only once. So, this rule is eliminated to produce the final result.

S ➔ B, B B ➔ 1, 2, 3

Note that the above grammar accepts only the sequence 123123.

III. IMPLEMENTATION OF PROPOSED SYSTEM

The implementation of proposed system is divided into these parts.

3.1 Disaster Site: First one establishment of site where are disaster may be occur or in other words they are which are identical for disaster occurrence.
Generate data through sensors and transfer data via network may be MANET.

3.2 Secure Communication:

For secure communication we implement ORDES approach with Fuzzy Error Correction Code
Now Consider Secure channel for secure communication and the main aspect in reference of unnatural disaster we use fuzzy error correction technique with ORDES Algorithm.

Now apply ORDES Algorithm on Generated Data for High Level Security.
In encryption phase, ORDES take output of compression phase as a message block M_n and a new generated key K_(new n) implement encryption process as per traditional DES.
In this process, New key is also make 16 different key for every round of ORDES using shifting property as per traditional DES. For every block of message M, new key K_(new n) makes a new key block for every round of DES to implement in the encryption process.
Decryption Process is the inverse step of encryption process. In decryption, we also use the same key which is used in encryption.

〖Ci=E〗_(K_(new n)) {mi} and 〖mi=D〗_(K_(new n))) {Ci},

where 1⩽ i ⩽ n.
Cipher Text C={C1,C2………Ci} and Plain Text M={m1,m2………mi}

Compression using SEQUITUR:
After quantization, the scheme uses a filter to pass only the string of non-zero coefficients. By the end of this process we will have a list of non-zero tokens for each block preceded by their count.
DCT based image compression using blocks of size 8x8 is considered. After this, the quantization of DCT coefficients of image blocks is carried out. The SEQUITER compression is then applied to the quantized DCT coefficients.

Error Correction:
Receiver check that , he will realize that there is an error occur during the transmission. Receiver apply the error correction function f to .

Then receiver will compute nearness

$$(t(c), f(c')) = dist(t(c) f(c'))/n$$

$$FUZZ(c') = 0 \qquad \text{if nearness}(c, c') = z \le z_0 < 1$$
$$= z \qquad \text{otherwise}$$

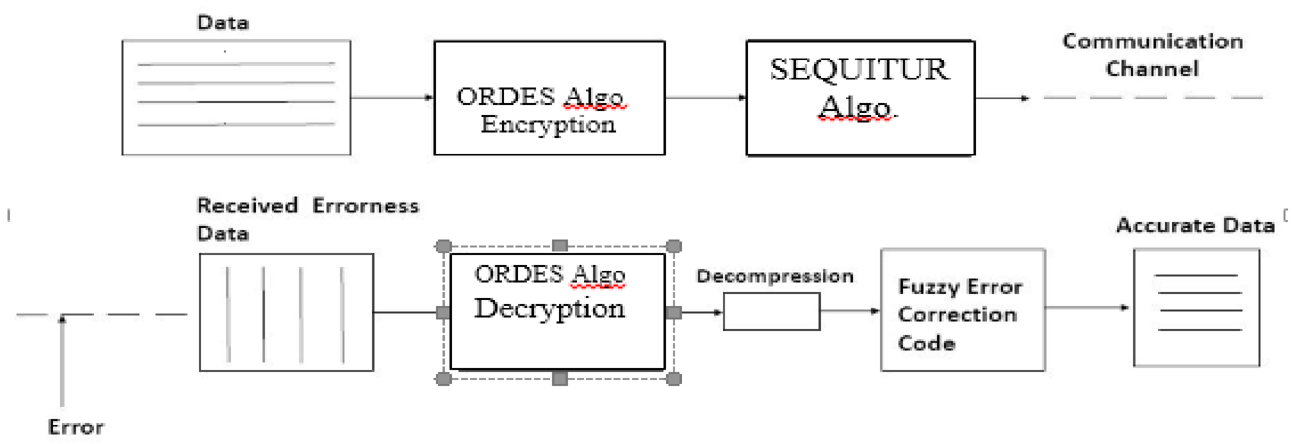

Fig 2. Architecture of disaster management system

IV. Conclusion

In this paper, we implement ORDES scheme, Sequitur and Fuzzy error correction code for Disaster Management. As we know that ORDES provides n-times more security in compare of DES but only in two cases ORDES works like DES.
Case I
If we take one and only one Key on the place of n keys then 〖K_new (x〗_p) is K, at this condition our approach works like DES.

Case II
If 〖K_new (x〗_1) = 〖K_new (x〗_2) = 〖K_new (x〗_n), than our approach also works like DES.

If, after encryption anyone wants to communicate message with commitment. The encoded message is transmitted. It is possible that during the transmission some bits of are changed. The receiver receives the incorrect message . He solves the decoding problem, that is, he calculates such that is minimum. If the error is not too big, that is, , where is the minimum distance of any two distinct code words, then is equal to the original message .

REFERENCES

[1] Data Encryption Standard, Federal Information Processing Standards Publication (FIPS PUB) 46, National Bureau of Standards, Washington, DC (1977).

[2] Eli Biham, Adi Shamir, Differential Cryptanalysis of DES-like Cryptosystems, Journal of Cryptology, Vol. 4 No. 1, Springer, pp. 3–72, 1991.

[3] Eli Biham, Adi Shamir, Differential Cryptanalysis of the Full 16-Round DES, Advances in Cryptology, proceedings of CRYPTO '92, Lecture Notes in Computer Science 740, Springer, 1993.

[4] Eli Biham, Adi Shamir, Differential Cryptanalysis of the Data Encryption Standard, Springer, 1993.

[5] David Chaum, Jan-Hendrik Evertse, Cryptanalysis of DES with a Reduced Number of Rounds: Sequences of Linear Factors in Block Ciphers, Advances in Cryptology, proceedings of CRYPTO '85, Lecture Notes in Computer Science 218, pp. 192–211, Springer, 1986.

[6] Donald W. Davies, Investigation of a Potential Weakness in the DES Algorithm, private communications, 1987.

[7] Donald W. Davies, Sean Murphy, Pairs and Triplets of DES S-Boxes, Journal of Cryptology, Vol. 8, No. 1, pp. 1–25, Springer, 1995.

[8] Eli Biham, Alex Biryukov, An Improvement of Davies' Attack on DES, Journal of Cryptology, Vol. 10, No. 3, pp. 195–206, Springer, 1997.

[9] Sebastien Kunz-Jacques, Frederic Muller, New Improvements of Davies-Murphy Cryptanalysis, Advances in Cryptology, proceedings of ASIACRYPT 2005, Lecture Notes in Computer Science 3788, pp. 425–442, Springer, 2005.

[10] Kunz-Jacques, S., Muller, F.: New Improvements of Davies-Murphy Cryptanalysis.In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 425–442. Springer, Heidelberg (2005)

[11] Mitsuru Matsui, Linear Cryptanalysis Method for DES Cipher, Advances in Cryptology, proceedings of EUROCRYPT '93, Lecture Notes in Computer Science 765, pp. 386–397, Springer, 1994.

[12] Takeshi Shimoyama, Toshinobu Kaneko, Quadratic Relation of S-box and Its Application to the Linear Attack of Full Round DES, Advances in Cryptology, proceedings of CRYPTO '98, Lecture Notes in Computer Science 1462, pp. 200–211, Springer, 1998.

[13] CNET News.com, Users take crack at 56-bit crypto. Available on-line at http://news.com.com/2100-1023-278658.html?legacy=cnet, 1997.

[14] RSA Data Security, Team of Universities, Companies and Individual Computer Users Linked over the Internet Crack RSA's 56-Bit DES Challenge. Available on-line at: http://www.rsasecurity.com/news/pr/970619-1.html, 1997.

[15] Electronic Frontier Foundation, Cracking DES, Secrets of Encryption Research, Wiretap Politics & Chip Design, O'reilly, 1998.

[16] . Nicolas T. Courtois, Gregory V. Bard, Algebraic Cryptanalysis of the Data Encryption Standard. Available on-line at: http://eprint.iacr.org/2006/402.pdf, 2006.

[17] M.Matsui: "The First Experimental Cryptanalysis of the Data Encryption Standard", Crypto'94, LNCS 839, Springer, pp. 1-11, 1994.

[18] Eli Biham and Adi Shamir: "Differential Cryptanalysis of DES-like Cryptosystems". Journal of Cryptology, vol. 4, pp. 3-72, IACR, 1991.

[19] M. Matsui: "Linear Cryptanalysis Method for DES Cipher",Eurocrypt'93", LNCS 765,Springer, pp. 386-397, 1993.

[20] Orr Dunkelman, Gautham Sekar, and Bart Preneel: "Improved Meet-in-the- Middle Attacks on Reduced-Round DES", To appear in Indocrypt 2007.

[21] Alejandro Hevia, Marcos Kiwi, "Strength of two data encryption standard implementation under timing attacks", ACM Transactions on Information and System Security (TISSEC), Volume 2, Issue 4 (November 1999) Pages: 416 – 437.

[22] M.Matsui: "The First Experimental Cryptanalysis of the Data Encryption Standard", Crypto'94, LNCS 839, Springer, pp. 1-11, 1994.

[23] Lars R. Knudsen, John E. Mathiassen, A Chosen-Plaintext Linear Attack on DES, proceedings of Fast Software Encryption 7, Lecture Notes in Computer Science 1978, pp. 262–272, Springer, 2001.

[24] Manuel Blum, "Coin flipping by telephone," Advances in Cryptology : A Report on CRYPTO'81,pp.11-15,1981.

[25] A.Juels and M.Wattenberg, " A fuzzy commitment scheme", In Proceedings of the 6th ACM Conference on Computer and Communication Security, pp.28-36, November 1999.

[26] Ramveer Singh and Deo Brat Ojha, " An Ordeal Random Data Encryption Scheme (ORDES)", In IJCA,2010 (Paper Acepted).

[27] Ramveer Singh and Deo Brat Ojha, " An Ordeal Randomized Secure Data Encryption Scheme (ORSDES) ", In IJCIM,2010 (Paper Acepted).

[28] V.Pless, " Introduction to theory of Error Correcting Codes", Wiley , New York 1982.

[29] A.A.Al-saggaf , H.S.Acharya, "A Fuzzy Commitment Scheme" IEEE International Conference on Advances in Computer Vision and Information Technology 28-30 November 2007 – India.

[30] Xavier Boyen. Reusable Cryptographic Fuzzy Extractors. In 11th ACM Conference on Computer and Communications Security (CCS 2004), pages 82-91. ACM Press, 2004.

[31] J.P.Pandey, D.B.Ojha, Ajay Sharma, "Enhance Fuzzy Commitment Scheme: An Approach For Post Quantum Cryptosystem", in Journal of Applied and Theoretical Information Technology, (pp 16-19) Vol. 9, No. 1, Nov. 2009.

[32] Ramveer Singh , Awakash Mishra and D.B.Ojha "An Instinctive Approach for Secure Communication – Enhanced Data Encryption Standard (EHDES)" International journal of computer science and Information technology, Sep. 2010 (Paper Acepted)

[33] D.B. Ojha, Ramveer Singh, Ajay Sharma, Awakash Mishra and Swati Garg "An Innovative Approach to Enhance the Security of Data Encryption Scheme" International Journal of Computer Theory and Engineering, Vol. 2,No. 3, June, 2010,1793-8201

[34] Nameer N. EL-Emam, "Hiding a Large Amount of Data with High Security Using Steganography Algorithm" Applied Computer Science Department, Faculty of Information Technology, Philadelphia University, Jordan

[35] Borie J., Puech W., and Dumas M., "Crypto-Compression System for Secure Transfer of Medical Images", 2nd International Conference on Advances in Medical Signal and Information Processing (MEDSIP 2004), September 2004.

[36] N.Walkinshaw, S.Afshan, P.McMinn "Using Compression Algorithms to Support the Comprehension of Program Traces" Proceedings of the International Workshop on Dynamic Analysis (WODA 2010) Trento, Italy, July 2010.