

**DATA SECURITY IN
FAULT TOLERANT HARD REAL TIME SYSTEMS -
USE OF TIME DEPENDANT MULTIPLE RANDOM CIPHER CODE**

A thesis submitted

by

VARGHESE PAUL

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

of

**COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY
UNDER FACULTY OF TECHNOLOGY**


**DEPARTMENT OF COMPUTER SCIENCE
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY
KOCHI – 682 022**

APRIL 2003

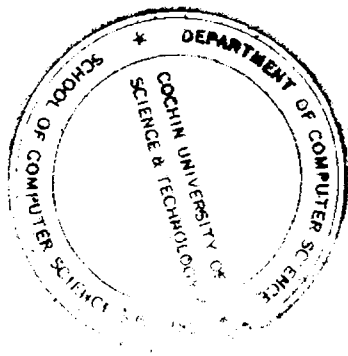
CERTIFICATE

This is to certify that the thesis entitled *DATA SECURITY IN FAULT TOLERANT HARD REAL TIME SYSTEMS – USE OF TIME DEPENDANT MULTIPLE RANDOM CIPHER CODE* is a report of the original work carried out by Mr. Varghese Paul, under my supervision and guidance in Department of Computer Science, Cochin University of Science and Technology. No part of the work reported in this thesis has been presented for any other degree from any other University.

Kochi 682 022
April 05, 2003



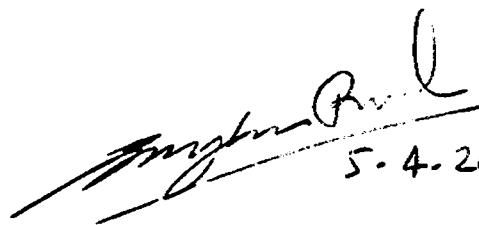
Dr.K Poullose Jacob
(Supervising Teacher)
Professor and Head
Department of Computer Science
Cochin University of Science and Technology



DECLARATION

I hereby declare that the work presented in this thesis is based on the original work done by me under the supervision of Dr. K Poulose Jacob, Professor and Head, Department of computer Science, Cochin University of Science and Technology. No part of this thesis has been presented for any other degree from any other institution.

Kochi 682 002
April 05, 2003



5-4-2003.

VARGHESE PAUL

ACKNOWLEDGEMENTS

I wish to place on record my profound sense of gratitude and thanks to Prof (Dr.) K Poulse Jacob, Head, Department of Computer Science for his inspiring guidance and constant encouragement throughout this investigation.

I am deeply indebted to Dr. S. Sasikumaran, Professor of Computer Applications, Institute of Management, Government of Kerala, Thiruvananthapuram for the guidance and supervision in the initial stage and valuable advice and unstinted support throughout the course of this work.

Dr. A. P. Kuriakose, Former Dean, Faculty of Technology, CUSAT, deserves special thanks for arranging the global contest for checking the vulnerability of TDMRC Code developed during this research work.

I thank Dr. Jacob Mathew, President, International Cyber Business Systems Inc, Cleveland, USA for the help and guidance rendered to complete this work.

My sincere thanks to the Librarian, Central Library, VSSC, Thiruvananthapuram for granting permission and facilitating reference of the books and journals there.

The help and assistance rendered by the teachers and staff of the department of Computer Science and School of Engineering are gratefully acknowledged.

VARGHESE PAUL

C O N T E N T S

CHAPTER I - INTRODUCTION	1
CHAPTER II - DATA SECURITY IN FAULT TOLERANT HARD REAL TIME SYSTEMS – REVIEW OF EARLIER WORK	
2.1 Introduction	7
2.2 Fault Tolerant Hard Real Time System Design	9
2.3 Data Security in Fault Tolerant Hard Real Time Systems	14
2.4 Eaves Dropping in the Communication Channels of FTHRT Systems	31
CHAPTER III - REVIEW OF EARLIER WORK IN CRYPTOGRAPHY	
3.1 Introduction	33
3.2 Information Security and Cryptography	35
3.3 Cryptographic Goals	37
3.4 Encryption Domains and Co domains	38
3.5 Encryption and Decryption Transformations	39
3.6 Symmetric-Key Encryption	42

3.7	Public Key Cryptography	43
3.8	Symmetric Key vs. Public Key Cryptography	46
3.8.1	Advantages of Symmetric Key Cryptography	47
3.8.2	Disadvantages of Symmetric Key Cryptography	48
3.8.3	Advantages of Public Key Cryptography	48
3.8.4	Disadvantages of Public Key Cryptography	49
3.9	Block Ciphers	51
3.9.1	Classical Ciphers and Historical Development	54
3.9.2	Data Encryption Standard	56
3.9.3	Fast Data Encipherment Algorithm	59
3.9.4	International Data Encryption Algorithm	61
3.9.5	Secure And Fast Encryption Algorithm	62
3.9.6	RC 5	63
3.9.7	Other Block Ciphers	64
3.10	Stream Ciphers	67
3.10.1	Stream Ciphers based on LFSRs	71
3.10.2	Other Stream Ciphers	71
3.11	Crypt Analysis	77
3.12	Digital Signatures	81
3.13	Authentication and Identification	82
3.14	Hash Functions	83

3.15	Protocols And Mechanisms	83
3.16	Key Establishment, Management and Certification	85
3.16.1	Trusted Third parties (TTP) and Public Key Certificates	86
3.17	Pseudorandom Numbers and Sequences	88

CHAPTER IV - TIME DEPENDANT MULTIPLE RANDOM CIPHER CODE

4.1	Introduction	92
4.2	Mandatory Requirement of Practical Encryption Systems	92
4.3	Structure of Time Dependant Multiple Random Cipher Code (TDMRC Code)	94
4.4	Key of TDMRC Code	95
4.5	Algorithm of TDMRC Code	98
4.6	Time Dependency of TDMRC Code	100
4.7	Poly Alphabetic Nature	102
4.8	Pseudo Random Nature	104
4.9	Variable Block Length	105
4.10	Comparison of TDMRC Code with other Conventional Schemes	106
4.11	Crypt Analysis of TDMRC Code	107
4.12	Vulnerability Checking of TDMRC Code	110

CHAPTER V - IMPLEMENTATION OF TDMRC CODE
IN FAULT TOLERANT HARD REAL TIME
SYSTEMS.

5.1	Introduction	114
5.2	Implementation of TDMRC Code in Fault Tolerant Hard Real Time System	115
5.3	Experimental Set up to Study the Implementation of TDMRC Code in FTHRT Systems.	116
5.4	Limitations when TDMRC Code is Implemented in Fault Tolerant Hard Real Time System	118

CHAPTER VI – CONCLUSIONS AND SUGGESTION FOR
FURTHER WORK

APPENDIX – I Global Contest Arranged for Checking the
Vulnerability of TDMRC Code

APPENDIX – II List of Institutions where Technical Talk on Data
Security and Cryptography was Given and TDMRC
Code was Introduced for Crypt Analysis Trial.

PUBLICATIONS

REFERENCES

CHAPTER - I

INTRODUCTION

Any system where a timely response by the computer to external stimuli is vital is a Real Time System. Real Time Systems must satisfy explicit response time constraints or risk severe consequences including failure. Logical correctness of the output of such systems is based on both the correctness of the inputs and their timeliness.

Real Time Systems are typically designed for specific applications. They have the advantage that the characteristics of the application and its environment are more precisely known compared to their general purpose counterpart. As a result it is possible to fine tune real time systems more precisely for optimum performance.

Real Time Systems can be classified into two categories – *Soft Real Time Systems and Hard Real Time Systems*. In Soft Real Time Systems performance is degraded but not destroyed by failure to meet response time constraints whereas in Hard Real Time systems failure to meet response time constraints will lead to failure of the system itself.

Distributed computer systems are increasingly being employed for critical applications, such as aircraft control, industrial process control,

and banking systems. Maximizing performance has been the conventional objective in the allocation of tasks for such systems. Inherently, distributed systems are more complex than centralized systems. The added complexity could increase the potential for system failures. Some work has been done in the past in allocating tasks to distributed systems, considering reliability as the objective function to be maximized. Reliability is defined to be the probability that none of the system components fails while processing. This, however, does not give any guarantees as to the behavior of the system when a failure occurs. A failure, not detected immediately, could lead to a catastrophe. Such systems are unsafe.

A system is considered fault tolerant if the behaviour of the system, despite the failure of some of its components, is consistent with its specifications. Fault tolerant systems have the capability to function in the presence of fault. By employing fault tolerance, many potential failures are averted, thereby increasing the reliability. Another goal of fault tolerance is to increase the system availability, that is, increase the time for which the system is available for user services. Redundant systems are used for achieving this quality. When redundant systems are used consistency of data among various systems is of prime importance. The available data and the processed output should be compared between various redundant systems at frequent intervals.

When the redundant systems are located at geographically distant places this comparison is to be done by transmitting data and output through communication links between various constituent systems. The rate of data transmission should also be high. These

communication channels are to be well protected against intruders especially when the system is used for strategic applications like military, aerospace research, nuclear research etc. Since Fault Tolerant Hard Real Time Systems are widely used in high tech warfare also, the chance of intrusion and risk of forced leakage of confidential information is very high in this field. To ensure correct data reception there exist many error checking and error correcting codes. But for security from eaves droppers it is better to use encryption techniques in this kind of network so that the actual information can be kept away from the intruders even if they manage to gain access to the communication channel.

The present research problem is to study the existing encryption methods and to develop a new technique which is performance wise superior to other existing techniques and at the same time can be very well incorporated in the communication channels of Fault Tolerant Hard Real time systems along with existing Error Checking / Error Correcting codes, so that the intention of eaves dropping can be defeated. There are many encryption methods available now. Each method has got it's own merits and demerits. Similarly, many crypt analysis techniques which adversaries use are also available.

Information on fault tolerant hard real time systems and encryption methods, were surveyed in research journals as well as books. The internet was often found as a good repository in collecting details.

Detailed study conducted on data encryption techniques lead to the development of a new data encryption method named *Time Dependant*

Multiple Random Cipher Code (TDMRC Code) which can be very effectively used for this purpose. This particular method has many complexities compared to other methods and cryptanalysis is practically impossible.

This method is a product code which uses variable block length where as the conventional methods are of fixed block length.

The code used for any particular character differs depending upon time – that is, coding is time dependant. Even for centi second difference, the codes will change.

The code used for the same character at different locations of the plain text are different – that is, coding is poly alphabetic. Also, Pseudo Random Number generation technique is used for code generation.

Vulnerability check of the proposed system was carried out during the course of the work. Also students, researchers and professionals were involved in the checking. A global contest with a reward was arranged to check the computational security and vulnerability of the proposed scheme. The details are given in Appendix - I

This thesis contains details of fault tolerant hard real time systems, various encryption systems in current practice, mandatory requirement of encryption methods, details of TDMRC Code and how to use this technique in communication channels linking redundant sub systems of fault tolerant real time systems.

The contents of each chapter are briefly described below.

Chapter I presents a brief description of the investigations carried out, highlighting the significance of the work. The methodology adopted and scope of the thesis is outlined.

Chapter II contains an overview of the relevant literature bringing out details of fault tolerant hard real time systems and existing security arrangements. Also limitations that exist with the current systems and the need for present day study has also been brought out.

Chapter III gives detailed review of existing encryption methods.. Also, merits and demerits of various techniques in this field are dealt with in this chapter.

Chapter IV contains details of Time Dependant Multiple Random Cipher Code. Time Dependant Multiple Random Cipher Code is a new technique of data encryption. This particular method has many complexities which make it more secure against crypt analysis. The various complexities are explained in detail in this chapter.

Chapter V deals with the implementation part of Time Dependant Multiple Random Cipher Code in Fault Tolerant Hard Real time system. Experimental set up and results of performance evaluation are described in this chapter.

Chapter VI summarises the conclusions drawn from the above investigations and discusses the scope for further work.

CHAPTER II

DATA SECURITY IN FAULT TOLERANT HARD REAL TIME SYSTEMS – REVIEW OF EARLIER WORK

- 2.1 Introduction
- 2.2 Fault Tolerant Hard Real Time System Design
- 2.3 Data Security in Fault Tolerant Hard Real Time Systems
- 2.4 Eaves Dropping in the Communication Channels of FTHRT Systems

2.1 Introduction

There are many phases that a system typically undergoes for supporting fault tolerance. These phases are error detection, damage confinement, error recovery, and fault treatment and continued service. Since error detection is the starting point of supporting fault tolerance, a fault tolerance strategy can be, at most, as good as its error detection method. Some of the common error detection methods are replication checks, timing checks, structural and coding checks, reasonableness checks, and damage checks.

As the error may be detected sometimes after the failure has occurred, the next step in supporting fault tolerance is to determine the extent of damage to the system state by the failure. This is done in the damage confinement phase. For damage assessment, interaction between different components will have to be examined because it is by interaction that errors can propagate. The goal is to identify some boundaries within which the spread of the error is confined. These boundaries can be dynamically determined after the error has been detected by examining the component interactions, or the component interaction can be constrained in such a manner that the error spread is limited to some predefined boundaries.

The next step is error recovery. Once the spread of an error has been identified, the error has to be removed from the system. This is done by error recovery. The two major techniques are backward error recovery and forward error recovery. In backward error recovery,

during normal computation the state of the system is periodically checkpointed. For recovery, the checkpointed state of the system is restored. If the failure is occurred after the checkpoint, this rollback will remove the error. In forward recovery, on the other hand, no previous system state is available. The goal is to make the system state error free by taking corrective actions. While backward recovery is a general technique, forward recovery requires a good diagnosis about the nature of the error.

The last phase is fault treatment and continued service. In the earlier phases, the focus was on error and error removal. But the root cause of any error is fault. Though in some cases, particularly with transient faults, just error recovery may suffice, in others, after error recovery, we must remove the fault that caused the error in order to avoid future failures. This is done in this phase. First the fault is located by identifying the faulty component. Then the system is repaired by reconfiguring the system by using the built in redundancy such that either the failed component is not used or is used in a different manner.

The availability of a system can be defined as

$$\text{MTBF} / (\text{MTBF} + \text{MTTR})$$

where MTBF is Mean Time Between Failures and
MTTR is Mean Time To Repair.

2.1 Fault Tolerant Hard Real Time System Design

[AVI 1977] presents an excellent review of the methodology of fault tolerant system design. Road blocks in fault tolerant computing are (i) lack of continuity – many of the techniques are never disclosed (trade secrecy) thus resulting in the repetition of many mistakes in the past (ii) lack of cost / benefit measures (iii) lack of specification and acceptance tests (iv) fragmentation of efforts (v) inertia in the design process (vi) resistance to potential impact – successful introduction of fault-tolerance may cause some de-emphasis of several currently flourishing activities.

A systematic methodology for the incorporation of fault-tolerance into the architecture of computing systems is presented in [AVI 1975]. Two approaches are, fault- tolerance and fault-intolerance. In the first approach, reliability is obtained by the use of protective redundancy for error detection and recovery, while in the second approach, reliability must be obtained by the priority for elimination of the causes of unreliability.

[WEN 1974] examines the reliability, availability, recovery time, data protection and maintainability requirements for five classes of computer applications (i) general purpose time shared (ii) general purpose batch (iii) communication (iv) super fast and (v) aerospace. Possible ways of introducing redundancy are given – starting from a system containing only byte error detection in many memory to a system containing uniform redundancy (i.e. where programs are run simultaneously on two computer units).

The STAR is a fault-tolerant computer primarily intended for use in hard real time application like spacecraft guidance, control and data acquisition systems on long unmanned space missions. [AVI 1971] explains the notable features of this computer as (i) use of special processor (TARP – Test And Repair Processor) to monitor the performance of the computer and to arrange recovery when detected an error, (ii) use of hybrid redundancy – STAR employed masking redundancy (triple modular redundancy) for the implementation of TARP and standby sparing redundancy for the other modules of the computer.

[AVI 1971] describes a hybrid-redundant multiprocessor organization for space applications. Each processing unit and memory unit is triplicated and there are number of spare units available to replace failed units.

In [MER 1976], a multiprocessor system for aerospace application is described. The system uses standby sparing redundancy for processors and memory units. The notable features of this system are (i) it is reconfigurable – processors and memory units can be removed or added dynamically (ii) the provision of an automatic rollback facility whereby the state of a computation can be restored to an earlier state, (iii) implementation of this rollback facility by hardware, and (iv) automatic generation of rollback facility by a programmer is not concerned with their specification

[HAM 1972] concentrates on hardware fault tolerance of applications such as telephone exchanges which use stored program control and requires a mean time between failures (unavailability exceeding 10 minutes) of 50 years. This kind of system consists of functionally equivalent processors connected to store and input-output modules. An important aspect of these processors is their use of a capability mechanism for the protection of information stored in memory modules. When a processor detects an error it generates a fault interrupt so that programmed error recovery may be initiated.

[HOP 1975] describes a multiprocessor system designed for use as a switching node in the ARPA network. The reliability goal was to construct a system that would survive not only transient failures but also solid failures of any single component. The hardware consists of buses joined together by special bus couplers allowing units on one bus to access those on another. The buses are of three kinds: (i) processor bus, each bus can contain two processors with local memory, (ii) memory bus, to house the segments of large shared memory, and (iii) I/O bus for device controllers. Hardware reliability is achieved by keeping sufficient extra copies of hardware resources and by ensuring that these hardware copies are isolated as much as possible (so that a failure of one unit should not affect others). The paper also describes the software strategies used for error detection and recovery.

[FISH 1973] describes design philosophy for multiprocessor systems intended for ultra reliable hard real time applications. The design conditions are (i) use of *off-the-shelf* components and subsystems, (ii) realistic cost constraints (i.e. only a limited use of hardware

redundancy), and (iii) dedicated application usage. The authors make two observations: (i) increased use of LSI circuits would make exhaustive testing of complex units infeasible, and (ii) in the case of software, it is common knowledge that the complexity of such programs also makes their exhaustive testing impractical. Thus, both the hardware-and software may contain undetected design faults. The design presented in [FISH 1973] is tolerant to both classes of undetected faults. The basic idea is to run three or more versions of the application software on a suitably designed multiprocessor system that is capable of checking for any discrepancy in the results.

There are many ways of introducing redundancy into computer systems. Mathematical modeling plays an important role in the selection of appropriate techniques for meeting the given reliability goal. The reliability of a system can be quite sensitive to even small variations in certain design parameters; mathematical models provide the understanding and insight into the nature of this sensitivity. [LYO 1962] presents a thorough mathematical analysis of the triple-modular redundancy (TMR) technique. A TMR configuration with perfect voting circuits is first analysed and then the effect of imperfect voters on the reliability is considered.

[MAT 1970] says standby sparing redundancy technique has gained widespread usage in the implementation of fault-tolerant computers since it offers several advantages over static redundancy techniques. Computers employing the standby spare redundancy technique often need a hard core module for error detection and recovery. This module must be ultra reliable since its failure would leave the system fault

intolerant. The authors propose a *hybrid redundancy* technique for the design of hard core modules. It consists of a TMR (or its generalised version – NMR) system with standby spares. A detailed mathematical analysis of such a 'hybrid redundant' system is presented to show that a significant improvement over NMR systems can be obtained.

The authors of [BOU 1971] present reliability equations for most of the well known redundancy techniques. These techniques include : (i) TMR, (ii) TMR with sparing (hybrid redundancy), (iii) NMR with sparing (hybrid redundancy), and (iv) standby sparing. The last technique needs the facility of error detection and automatic reconfiguration (replacement of the failed component by one of the spares). Hence the authors introduce the important, notion of coverage, defined to be the conditional probability. A comparison of TMR and standby sparing is performed which indicates that TMR is almost unbeatable for short missions.

In [MAT 1975], the authors have developed a generalised reliability model (named GMR: General Modular Redundancy) such that the different redundancy techniques become particular cases of the model. It is therefore possible to present a unified treatment of reliability modeling. The advantage of this approach is that several different redundancy techniques can be compared with relative ease.

In [BOR 1974], a reliability model of PRIME is developed. A *crash* is defined as an interruption in the availability of a predefined minimum amount of computing power for a period of time exceeding the system's automatic recovery time. Four distinct causes of crashes

are assumed: (i) time domain multiple faults - crash due to a fault while recovering from an earlier fault, (ii) resource exhaustion - not enough resource units left to provide an acceptable service, (iii) space domain multiple faults - a crash due to the inadequacy of fault detection and recovery mechanisms, and (iv) solitary faults - the inability of the system to recover from a single fault

According to [SOM 1997], a good fault-tolerant system design requires a careful study of design, failures, causes of failures, and system response to failures. Planning to avoid failures is the most important aspect of fault tolerance. A designer must analyze the environment and determine the failures that must be tolerated to achieve the desired level of reliability. To optimize fault tolerance, it is important to estimate actual failure rates for each possible failure. The basic principle of fault-tolerant design is redundancy and there are three basic techniques to achieve it, namely, spatial (redundant hardware), informational (redundant data structures), and temporal (redundant computation).

2.2 Data Security in Fault Tolerant Hard Real Time System.

One of the essential properties a reliable system must possess is that of error confinement: the property of preventing an erroneous or corrupted software module from damaging other modules. Of equal importance is the requirement that the information stored in the system be secure from unauthorised access.

[WIL 1972] contains a concise and very reliable account of protection in computer systems. The chapter on memory management describes the two well known protection schemes: access list based and capability based. A discussion on hardware features necessary to support these schemes is also included. Later chapters describe user authentication mechanisms and file protection techniques. This book also contains details of file recovery techniques and methods of system restart after a failure.

[SAL 1975] gives a very comprehensive survey of techniques for protecting computer-stored information from unauthorised use or modification. Eight design principles for designing a protection system are given: (i) economy of mechanism. (ii) failsafe defaults (iii) complete mediation (iv) open design (v) separation of privilege (vi) least privilege (vii) least common mechanism, and (viii) psychological acceptability.

[POP 1975] describes the work undertaken at the University of California at Los Angeles with the aim of building a kernel for multi-user operating systems. The special feature of the kernel is that it is intended to provide a provably secure environment for information. The basic security is achieved by the creation of isolated virtual machines – the isolation guaranteeing the error confinement property. Great care has been taken to keep the security kernel as small and simple as possible so as to make the task of proving its correctness manageable.

[DEN 1976] develops an *information flow* model which can be used to specify secure information flow requirements. Some existing security systems are described using this model. It is shown that practical systems will need both access control and flow control to satisfy all security requirements.

[HAI 1976] discusses information protection in data bases. Protection problems in data bases are a great deal more complicated than the corresponding problems in operating systems. In the access matrix model of Lampson, it is assumed that if I wants to access an object j , then it is only necessary to check the access rights associated with the entry $A [I, J]$ and not any other entry, say $A [I, K]$. However, in the context of data bases, this form of checking is rather primitive. This is because the information present in a data base system must be considered as a collection of semantically inter-connected data items. This means that by accessing a given item, a user can implicitly gain some knowledge of other semantically connected items. The protection mechanism, however, must protect the data despite these connections.

Parity prediction arithmetic operators are compatible with systems checked by parity codes, however, they are not secure against single faults [NIC 1997]. This paper determines the necessary conditions for fault secureness and derives designs embodying these conditions.

In [KIM 1996] an instruction-retry policy is proposed to enhance the fault-tolerance of Triple Modular Redundant (TMR) controller computers by adding time redundancy to them. A TMR failure is said

to occur if a TMR system fails to establish a majority among its modules' outputs due to multiple faulty modules or a faulty voter.

An adaptive computing system is one that modifies its behavior based on changes in the environment. Since sites connected by a local-area network inherently have to deal with network congestion and the failure of other sites, distributed systems can be viewed as an important subclass of adaptive systems. As such, use of adaptive methods in this context has the same potential advantages of improved efficiency and structural simplicity as for adaptive systems in general. [HIL 1996] describes a model for adaptive systems that can be applied in many scenarios arising in distributed and fault-tolerant systems. This model divides the adaptation process into three different phases - change detection, agreement, and action - that can be used to describe existing algorithms that deal with change, as well as to develop new adaptive algorithms.

A fault-tolerant data transmission model based on the redundant residue number system is proposed in [YAN 1996]. It can transmit data correctly between two ends unless the residue errors exceed the error-correcting capability.

Distributed voting is an important problem in reliable computing. In an N Modular Redundant (NMR) system, the N computational modules execute identical tasks and they need to periodically vote on their current states. [LIH 1998] proposes a deterministic majority voting algorithm for NMR systems. The proposed algorithm uses error-

correcting codes to drastically reduce the average case communication complexity.

The delivery delay in a point-to-point packet switching network is difficult to control due to the congestion among randomly-arriving packets at each node. Despite this difficulty, there are an increasing number of applications that require packets to be delivered reliably within pre specified delay bounds. [ZHE 1998] shows how this can be achieved by using real-time channels which make *soft* reservation of network resources to ensure the timely delivery of real-time packets.

[SRI 1999] describes a method to determine an allocation that introduces safety into a heterogeneous distributed system and at the same time attempts to maximize its reliability.

[BAL 1999] presents an index-based checkpointing algorithm for distributed systems with the aim of reducing the total number of checkpoints while ensuring that each checkpoint belongs to at least one consistent global checkpoint (or recovery line). The algorithm is based on an equivalence relation defined between pairs of successive checkpoints of a process which allows, in some cases, to advance the recovery line of the computation without forcing checkpoints in other processes. The algorithm is well-suited for autonomous and heterogeneous environments, where each process does not know any private information about other processes and private information of the same type of distinct processes is not related.

[ROM 1998] presents a recovery block (RB) scheme that is suitable for a real time application which has a predictable fault tolerant behaviour. The basic problem to be tackled for introducing the RB scheme is analysed in real time systems and propose some approaches and solutions allowing to handle them. The differences between using RBs for hard real time tasks and for soft ones are considered. The computational and timing model of the RB execution is described. These RBs can be used as building blocks (pads of tasks) for designing real time systems with predictable behaviour.

[BER 1998] describes 2 models for communication employing recovery blocks. Model #1 considers 2 Recovery Blocks, RB-1 & RB-2, where RB-2 receives some data from RB-1. Thus, if a version in RB-2 fails then RB-1 has to rollback to its initial state. Model #2 considers 2 Recovery Blocks in conversation: both blocks must satisfy their respective acceptance tests before any of the blocks are allowed to exit from the conversation.

Advanced satellites with on-board base-band switching processors have Time-Space-Time (T-S-T) structures which are similar to the terrestrial switching networks (Sw-Nw). Generally, the satellite systems require higher reliability than ground equipment because of more severe environment and lack of repair. [KAN 1996] proposes fault-tolerant satellite on-board T-S-T Sw-Nw with multiple separated space switches instead of a single space switch. Mean time to unreliable operation (MTUO) is treated as a performance & reliability index for the T-S-T systems with multiple separated space switches as well as conventional T-S-T systems. The MTUO vary depending on

the threshold level of blocking-probability and the offered traffic. In general, T-S-T Sw-Nw with multiple space switches have better performance and reliability than those with the single space switch.

[CHI 1996] proposes a new approach for implementing rollback-recovery in a distributed computing system. A concept of logical ring is introduced for the maintenance of information required for consistent recovery from a system crash. Message processing order of a process is kept by all other processes on its logical ring. Transmission of data messages are accompanied by the circulation of the associated order message on the ring. The sizes of the order messages are small. In addition, redundant transmission of order information is avoided, thereby reducing the communication overhead incurred during failure-free operation. Furthermore, updating of the order information and garbage collection task are simplified in the proposed mechanism.

The TCP/IP network protocol has gained wide acceptance as the de facto standard for inter-system data transportation in the manufacturing automation environment. While it was designed to be a fault-tolerant, robust protocol family, TCP/IP is still susceptible to many forms of deliberate and accidental attack which can compromise data integrity and network effectiveness. [RAY 1997] addresses ways in which the impact of certain common security problems on a manufacturing automation network may be minimized.

Users are increasingly deploying high-speed networks in distributed computer systems. These networks may have stringent real-time and

fault-tolerance requirements [CHE 1997]. Fiber distributed data interface (FDDI) is a 100 Mbps local area network based on a token ring media access control protocol defined by the American National Standards Institute (ANSI) and Open System Interconnection (OSI) standards. It has built-in provisions for fault-tolerance and real-time communications.

In [GHO 1997], study of a scheme that provides fault-tolerance through scheduling in real-time multiprocessor systems is done. Multiple copies of dynamic, periodic, non preemptive tasks are used in the system, and deallocation and overloading is done to achieve high acceptance ratio. [GHO 1997] compares the performance of fault-tolerant scheduling schemes, and determines how much each of deallocation and overloading affects the acceptance ratio of tasks.

The bound on component failures and their spatial distribution govern the fault tolerance of any candidate error-detecting algorithm. For distributed memory multiprocessors, the specific algorithm and the topology of the processor interconnection network define these bounds [SCH 1997].

[TAK 1996] presents field data from the Hiten satellite On Board Computer (OBC) which was launched on 1990 January 24, and completed its mission on 1993 April 10. The components in the OBC experienced 655 Single Event Upsets (SEU) caused by cosmic rays; the bursts of SEU were observed after 9 major solar flares. In spite of these SEU, the OBC worked correctly during the mission time, due to

the fault tolerance techniques. The field data reveal a statistical correlation between SEU and solar activities.

A fail-silent node is a self-checking node that either functions correctly or stops functioning after an internal failure is detected. Such a node can be constructed from a number of conventional processors. In a software-implemented fail-silent node, the nonfaulty processors of the node need to execute message order and comparison protocols to 'keep in step' and check each other, respectively [BRA 1996].

A simultaneous Fault Detection and Diagnostics (FDD) and Fault Tolerant Control (FTC) strategy for nonlinear stochastic systems in closed loops based on a Continuous Stirred Tank Reactor (CSTR) is presented in [ZHO 1998]. The purpose of control is to track the reactant concentration set point. Instead of output feedback, use Proportional Integral Derivative (PID) state feedback, which is essential to achieve FTC against sensor faults in proposed system.

[LUB 1998] describes a fault-tolerant system that is based on two replicas of a self-checking module and on an error-masking interface. The main contributions of this work rely on the strong-fail-safe design of the error-masking interface, and on the analysis of the competitiveness of this fault-tolerant scheme with respect to its reliability.

Structural Fault Tolerance (SFT) is the ability of a multiprocessor to reconfigure around a faulty processor or link in order to preserve its original processor interconnection structure. New (SFT)

multiprocessors should have to have a low switch and link overheads, but can tolerate a very large number of processor faults on the average [DUT 1997].

Check pointing enables to reduce the time to recover from a fault by saving intermediate results of the program in a reliable storage [ZIV 1997]. The length of the intervals between checkpoints affects the execution time of programs. On one hand, long intervals lead to long reprocessing time, while, on the other hand, too frequent check pointing leads to high check pointing overhead. [ZIV 1997] presents an on-line algorithm for placement of checkpoints. The algorithm uses knowledge of the current cost of a checkpoint when it decides whether or not to place a checkpoint. The total overhead of the execution time when the proposed algorithm used is smaller than the overhead when fixed intervals are used. Although the proposed algorithm uses only on-line knowledge about the cost of check pointing, its behavior is close to the off-line optimal algorithm that uses a complete knowledge of check pointing cost.

The early error detection and the understanding of the nature and conditions of an error occurrence can be useful to make an effective and efficient recovery in distributed systems. Various distributed system extensions were introduced for the implementation of fault tolerance in distributed software systems. These extensions rely mainly on the exchange of contextual information appended to every transmitted application specific message. Ideally, this information should be used for check pointing, error detection, diagnosis and recovery, should a transient failure occur later during the distributed

program execution. [SAL 1998] presents a generalized extension suitable for fault-tolerant distributed systems such as communication software systems and its detection capabilities are shown. The extension is based on the execution of message validity test prior to the transmission of messages and the piggybacking of contextual information to facilitate the detection and diagnosis of transient faults in the distributed system.

Reed-Solomon codes may be used to provide error correction for multiple failures. [PLA 1997] presents a complete specification of the coding algorithm plus details on how it may be implemented.

Fail-safety is a system attribute which ensures that a program either completes its execution satisfying its post-conditions in the normal manner or signals its failure to do so to its operating environment. Such an attribute is desirable of any system as it ensures the correctness of results which are produced. A very few modern sequential programming languages offer program fail-safety through the judicious use of a well designed exception handling mechanism. In [DRE 1996], the exception handling techniques that can be used in sequential systems are developed to provide the guidelines for fail-safe concurrent system design.

[HAN 1998] presents a scheme for restoring real-time channels, each with guaranteed timeliness, from component failures in networks. To ensure fast/guaranteed recovery, backup channels are set up, in addition to each primary channel. That is, a dependable real-time connection consists of a primary channel and one or more backup

channels. If a primary channel fails, one of its backup channels is activated to become a new primary channel. [HAN 1998] proposes a protocol which provides an integrated solution for dependable real-time communication in networks.

A critical problem in the design of ultra-reliable fault tolerant systems is that of how to bring a redundant member back on-line, after a transient fault, without degrading critical real-time functions. [SIM 1997] describes a hardware assisted recovery technique which uses memory tags to determine which memory segments need to be restored such that recovery can be performed incrementally without affecting real-time operational tasks.

A new generation of highly dependable real-time control systems (such as automotive brake-by-wire and steer-by-wire) is under development. Specific application domain requirements lead to the new features to be supported by the system software. These requirements are best supported by a time-triggered approach. Motorola is working on the time-triggered fault-tolerant communication hardware as well as participates in a software standardization committee. [DOR 2001] covers back-end system software for highly dependable real-time control systems including Operating System, Fault-Tolerant Communication Layer and Node-Local Configuration Tools

A condition monitoring system tracks real-world variables and alerts users when a predefined condition becomes true, e.g., when stock price drops, or when a nuclear reactor overheats. Replication of monitoring

servers can reduce the probability that an important alert is missed. However, replicated independent servers can sometimes report *conflicting* alerts to the user, causing confusion. [HUA 2001] identify and formally define three desirable properties of a replicated system, namely, orderedness, consistency, and completeness. It also proposes new monitoring algorithms that enforce some or all of the desired properties in different scenarios.

Modern systems such as nuclear power plants, the Space Shuttle or the International Space Station are examples of mission critical systems that need to be monitored round the clock. Such systems typically consist of embedded sensors in networked subsystems that can transmit data to central (or remote) monitoring stations [DEB 2001]. Qualtech Systems employs a Remote Diagnosis Server (RDS) to implement a remote health monitoring systems based on telemetry data from such systems. RDS can also be used to provide online monitoring of sensor-rich, network capable, legacy systems such as jet engines, building heating-ventilation-air-conditioning systems, and automobiles. The International Space Station utilizes a highly redundant, fault tolerant, software configurable, complex, bus system that links all major sub-systems. All sensor and monitoring information is communicated using this bus and sent to the ground station via telemetry. It is, therefore, a critical system and any failures in the bus system need to be diagnosed promptly.

On-line fault accommodation control problems under catastrophic system failures are investigated in [YEN 2001]. The main interest is focused on dealing with the unanticipated system component failures

in the most general formulation. A complete architecture of fault diagnosis and accommodation has also been presented by incorporating the developed intelligent fault tolerant control scheme with a cost-effective fault detection scheme and a multiple-model based failure diagnosis process to efficiently handle the false alarms and the accommodation of both the anticipated and unanticipated failures in on-line situations.

[BAJ 2001] considers the problem of designing fault tolerant control for transient failures in the flight control system caused by harsh electromagnetic environments and proposes an integrated local supervisory control of these systems. Sample design of a control mixer to achieve fault tolerance in the event of failures in the actuators is given.

In most communication networks, pairs of processors communicate by sending messages over a path connecting them. [HER 2001] presents communication-efficient protocols that quickly detect and locate any failure along the path. Whenever there is excessive delay in forwarding messages along the path, the protocols detect a failure (even when the delay is caused by maliciously programmed processors). The protocols ensure optimal time for either message delivery or failure detection.

Application Specific Programmable Processors (ASPP) provide efficient implementation for any number of specified functionalities. Due to their flexibility and convenient performance-cost trade-offs, ASPPs are being developed by DSP, video, multimedia, and embedded

IC manufacturers. [KAR 2000] presents two low-cost approaches to permanent fault tolerance of ASPPs. ASPP fault tolerance constraints are incorporated during scheduling, allocation, and assignment phases of behavioral synthesis. The first ASPP fault tolerance technique minimizes the hardware resources while guaranteeing that the ASPP remains operational in the presence of all unit faults. On the other hand, the second fault tolerance technique maximizes the ASPP fault tolerance subject to constraints on the hardware resources. These ASPP fault tolerance techniques impose several unique tasks, such as fault-tolerant scheduling, hardware allocation, and application-to-faulty-unit assignment. Phase clocks are synchronization tools that implement a form of logical time in distributed systems. For systems tolerating transient faults by self-repair of damaged data, phase clocks can enable reasoning about the progress of distributed repair procedures. This paper presents a phase clock algorithm suited to the model of transient memory faults in asynchronous systems with read/write registers.

In a distributed computing environment, exceptions may be raised simultaneously in different processing nodes and thus need to be treated in a coordinated manner. Mishandling concurrent exceptions can lead to catastrophic consequences. In [XUJ 2000] two kinds of concurrency are considered (i) several objects are designed collectively and invoked concurrently to achieve a global goal and (ii) multiple objects (or object groups) that are designed independently compete for the same system resources.

The fault injection approach presented in [CON 2000] can be used for validation of any fault-tolerant or highly available computing system. Intel Corporation developed the Teraflops supercomputer for the US Department of Energy (DOE) as part of the Accelerated Strategic Computing Initiative (ASCI). This was the most powerful computing machine available at that time, performing over two trillion floating point operations per second with the aid of more than 9,000 Intel processors. The Teraflops machine employs complex hardware and software fault/error handling mechanisms for complying with DOE's reliability requirements. [CON 2000] gives a brief description of the system architecture and presents the validation of the fault tolerance mechanisms.

For any fault-tolerant control method to take effect, sufficient redundancy must exist in the plant (process) to be controlled. [WUN 2000] establishes a means of measuring the level of redundancy in connection with feedback control by borrowing the notion of the second-order modes. In particular, it is assumed that foreseeable faults of a process are parameterized in the model of the process. The smallest second-order mode is used as a measure of the potentiality of the process to maintain a certain performance through controlled reconfiguration at the occurrence of the worst faults over a prescribed set in the fault parameter space. This measure is called by the authors as Controlled Reconfigurability. The Controlled Reconfigurability is calculated for two process models to show its relevance to redundant actuating capabilities in the models.

[APO 2000] addresses the topic of fault management proposing an extended information model, enabling to incorporate directly time related information as an attribute of the network state information. The proposal is based on a Temporal Management Information Base (TMIB) for fault management that represents the evolution in time of network resources. The proposed model can be implemented in a centralized or distributed environment. Moreover, the necessary primitive fault management services are defined.

Information on Fault Tolerance Latency (FTL), which is defined as the total time required by all sequential steps taken to recover from an error, is important in the design and evaluation of fault-tolerant computers used in safety-critical real-time control systems with deadline information.

[CHA 1999A] describes an approach to using Commercial-off-the-Shelf (COTS) products in highly reliable systems. The methodology calls for multi-level fault-protection. The methodology realizes that COTS products are often not developed with high reliability in mind. Nevertheless, by using multi-level fault protection, the same level of reliability as the traditional full-custom fault tolerance approach can be achieved. A low-cost and fast Totally Self-Checking (TSC) checker for m-out-of-n code, is presented in [CHA 1999]

Today's aircrafts use ultra-reliable real time controls for demanding functions such as Fly-By-Wire (FBW) flight control. Future aircraft, spacecraft and other vehicles will require use of these types of control for functions that currently are allowed to fail, fail to degraded

operation, or require human intervention in response to failure. The use of low-cost sensors with digital outputs, digitally commanded fault-tolerant actuation devices and interconnecting networks of low-cost data buses which offers more affordable ultra-reliable systems are presented in [HAM 1972].

2.3 Eves Dropping in the Communication Channels of FTHRT Systems

The error checking and correcting methods seen in the above discussions are helpful in ensuring the correctness of data. Since FTHRT systems are mainly used for critical and very secured applications, it's communication channels are to be protected against eaves dropping also. Data received intact at the receiving end doesn't guarantee that it is secured data.

During the literature survey no journal or publication giving details about any techniques used in FTHRT system to block eaves dropping is come across. Since the FTHRT systems are used for very critical and sensitive applications, the full details of the security arrangements in the communication channel may not be published.

The research problem in this work is to develop a system which will guarantee that eaves dropping can be defeated in FTHRT systems. Very fast encryption method is proposed. For this the existing encryption schemes are studied and a suitable one is designed and proposed.

CHAPTER III

REVIEW OF EARLIER WORK IN CRYPTOGRAPHY

- 3.1 Introduction
- 3.2 Information Security and Cryptography
- 3.3 Cryptographic Goals
- 3.4 Encryption Domains and Co domains
- 3.5 Encryption and Decryption Transformations
- 3.6 Symmetric Key Encryption
- 3.7 Public Key Cryptography
- 3.8 Symmetric Key vs. Public Key Cryptography
- 3.9 Block Ciphers
- 3.10 Stream Ciphers
- 3.11 Crypt Analysis
- 3.12 Digital Signatures
- 3.13 Authentication and Identification
- 3.14 Hash Functions
- 3.15 Protocols and Mechanisms
- 3.16 Key Establishment, Management and Certification
- 3.17 Pseudorandom Numbers and Sequences

3.1 Introduction

Information security manifests itself in many ways according to the situation and requirement. Regardless of who is involved, to one degree or another, all parties to a transaction must have confidence that certain objectives associated with information security have been met.

Over the centuries, an elaborate set of protocols and mechanisms have been created to deal with information security issues when the information is conveyed by physical documents. Often the objectives of information security cannot solely be achieved through mathematical algorithms and protocols alone, but require procedural techniques and abundance of laws to achieve the desired result.

Conceptually, the way information is recorded has not changed dramatically over time. Whereas information was typically stored and transmitted on paper, much of it now resides on magnetic media and is transmitted via telecommunications systems, some wireless. What has changed dramatically is the ability to copy and alter information. One can make thousands of identical copies of a piece of information stored electronically and each is indistinguishable from the original. With information on paper, this is much more difficult.

What is needed then for a society where information is mostly stored and transmitted in electronic form is a means to ensure information security which is independent of the physical medium of recording or conveying it and such that the objectives of information security rely solely on digital information itself.

One of the fundamental tools used in information security is the signature. It is a building block for many other services such as non-repudiation, data origin authentication, identification, and witnessing, to mention a few.

Having learned the basics in writing, an individual is taught how to produce a handwritten signature for the purpose of identification. At contract stage the signature evolves to take a very integral part of the person's identity. This signature is intended to be unique to the individual and serve as a means to identify, authorize and validate.

With electronic information the concept of a signature needs to be redressed, it cannot simply be something unique to the signer and independent of the information signed. Electronic replication of it is so simple that appending a signature to a document not signed by the originator is almost a triviality.

For dealing this in electronic format analogues of the paper protocols currently in use are required. Hopefully these new electronic based protocols are at least as good as those they replace. There is a unique opportunity for society to introduce new and more efficient ways of ensuring information security. Much can be learned from the evolution of the paper based system, mimicking those aspects which have served us well and removing the inefficiencies.

Achieving information security in an electronic society requires a vast array of technical and legal skills. There is, however, no guarantee that

all of the information security objectives deemed necessary can be adequately met. The technical means is provided through cryptography.

3.2 Information Security and Cryptography

Cryptography has a long and fascinating history. The most complete non-technical account of the subject is Kahn's *The Code Breakers* [KAH 1967] which traces cryptography from its initial and limited use by the Egyptians some 4000 years ago, to the twentieth century where it played a crucial role in the outcome of both world wars. Completed in 1963, Kahn's book covers those aspects of the history which were most significant (up to that time) to the development of the subject.

[FEI 1973] provides an early exposition of block cipher ideas. The predominant practitioners of the art were those associated with the military, the diplomatic service and government in general. Cryptography was used as a tool to protect national secrets and strategies.

A concise and elegant way to describe cryptography given by Rivest in *Cryptography* [RIV 1990] is about communications in the presence of adversaries.

Meyer and Matyas [MEY 1982] say that the handwritten signature came into the British legal system in the seventeenth century as a means to provide various functions associated with information security. This book considers cryptography as it applies to information in digital form.

Beker and Pipe [BEK 1982] provides an introduction to the encryption of analogue signals, in particular, speech. Although in many cases physical means are employed to facilitate privacy, cryptography plays the major role. Physical means of providing privacy include fiber optic communication links, spread spectrum technology, and tamper resistant hardware.

Steganography is that branch of information privacy which attempts to obscure the existence of data through such devices as invisible inks, secret compartments, the use of subliminal channels, and the like. Kahn [KAH 1967] provides a historical account of various steganographic techniques also.

The proliferation of computers and communications systems in the 1960s brought with it a demand from the private sector for means to protect information in digital form and to provide security services. Beginning with the work of Feistel at IBM in the early 1970s and culminating in 1977 with the adoption as a U.S. Federal Information Processing Standard for encrypting unclassified information, DES, the Data Encryption Standard, is the most well-known cryptographic mechanism in history. The original specification of DES is the U.S. Federal Information Processing Standards Publication 46 [FIP 1977]. It remains the standard means for securing electronic commerce for many financial institutions around the world.

The most striking development in the history of cryptography came in 1976 when Diffie and Hellman published *New Directions in*

Cryptography [DIF 1976]. This paper introduced the revolutionary concept of public-key cryptography and also provided a new and ingenious method for key exchange, the security of which is based on the intractability of the discrete logarithm problem. Although the authors had no practical realization of a public-key encryption scheme at that time, the idea was clear and it generated extensive interest and activity in the cryptographic community.

3.3 Cryptographic Goals

The following security objectives form a framework upon which the others will be derived.

- (1) privacy or confidentiality
- (2) data integrity
- (3) authentication and
- (4) non-repudiation

A fundamental goal of cryptography is to adequately address these four areas in both theory and practice. Cryptography is about the prevention and detection of cheating and other malicious activities.

Cryptography, over the ages, has been an art practised by many who have devised ad hoc techniques to meet some of the information security requirements. The last twenty five years have been a period of transition as the discipline moved from an art to a science. There are now several international scientific conferences devoted exclusively to cryptography and also an international scientific organization, the

International Association for Cryptologic Research (IACR), aimed at fostering research in the area.

Trapdoor One-way functions were introduced by Diffie and Hellman [DIF 1976]. Merkle [MER 1979] describes it as a means to obtain public-key encryption schemes.

The basic concepts of cryptography are treated quite differently by various authors, some being more technical than others. Brassard [BRS 1988] provides a concise, lucid, and technically accurate account. Schneier [SCH 1996] gives a less technical but very accessible introduction.

Salomaa [SAL 1990], Stinson [STI 1995], and Rivest [RIV 1990] present more mathematical approaches. The comparison of an encryption scheme to a resettable combination lock is from Diffie and Hellman [DIF 1979].

Kerckhoffs' desiderata [KER 1883] was originally stated in French. Translation is given in Kahn [KAH 1967]. Shannon [SHA 1949] also gives desiderata for encryption schemes.

3.4 Encryption Domains and Codomains

A denotes a finite set called the *alphabet of definition*. For example, $A = \{0,1\}$, the *binary alphabet*, is a frequently used alphabet of definition. Note that any alphabet can be encoded in terms of the

binary alphabet. For example, since there are 32 binary strings of length five, each letter of the English alphabet can be assigned a unique binary string of length five.

M denotes a set called the *message space*. M consists of strings of symbols from an alphabet of definition. An element of M is called a *plaintext message* or simply a *plaintext*. For example, M may consist of binary strings, English text, computer code, etc.

C denotes a set called the *ciphertext space*. C consists of strings of symbols from an alphabet of definition, which may differ from the alphabet of definition for M . An element of C is called a *ciphertext*.

3.5 Encryption and decryption transformations

K denotes a set called the *key space*. An element of K is called a *key*.

Each element $e \in K$ uniquely determines a bijection from M to C , denoted by E_e is called an *encryption function* or an *encryption transformation*. E_e must be a bijection if the process is to be reversed and a unique plaintext message recovered for each distinct ciphertext.

For each $d \in K$, D_d denotes a bijection from C to M then D_d is called a *decryption function* or *decryption transformation*.

The process of applying the transformation E_e to a message $m \in M$ is usually referred to as *encrypting m* or the *encryption of m* .

ranged from 0.989 to 3.026 ppm and 1.177 to 2.809 ppm, respectively. Pb levels in *Bembrops*, *Uranoscopidae* and *Bothus* sp. separated from the mantle cavity of *Ancistrocheirus* spp. ranged from 0.855 to 2.047, 0.432 to 0.683 and 0.634 to 1.522 ppm, respectively. The essential metals Cu and Zn levels in the crustaceans and fishes were low except for a higher mean value of 16.859 ppm in *Heterocarpus gibbosus*. Elevated levels of Cr was noted in *Plesionika ensis* and Cr content ranged from 1.408 to 5.468 ppm and 20% of the samples had Cr content above 2ppm. Nevertheless, low content of Cr was found in *Heterocarpus gibbosus* (Table 6.2). Interestingly, Ni levels in the crustaceans and fishes separated from the mantle cavity of the oceanic squid were comparatively lower, with the highest value of 1.023 ppm found in *Plesionika ensis* and lowest value of 0.121 ppm noted in *Bothus* sp.

6.3.3. Metal levels in fishes collected in the same habitat area as neretic squids

The most abundant and recurrently occurring fishes along with neretic squids were *Priacanthus hamrur*, *Dactyloptena orientalis*, *Epinephelus diacanthus*, *Saurida tumbil*, *Upeneus* sp., *Alectus indica* and *Lutjanus lutjanus*. Trace metal distribution pattern in these fishes are presented in Table (6.3). Among the various fishes analysed Cd content in *Saurida tumbil* was comparatively higher (3.784 ± 3.499 ppm). The highest value recorded was 6.854 ppm. In *Upeneus* sp. mean Cd content was in the range of 0 to 0.876 ppm. Mean Pb levels were < 1

binary alphabet. For example, since there are 32 binary strings of length five, each letter of the English alphabet can be assigned a unique binary string of length five.

M denotes a set called the *message space*. M consists of strings of symbols from an alphabet of definition. An element of M is called a *plaintext message* or simply a *plaintext*. For example, M may consist of binary strings, English text, computer code, etc.

C denotes a set called the *ciphertext space*. C consists of strings of symbols from an alphabet of definition, which may differ from the alphabet of definition for M . An element of C is called a *ciphertext*.

3.5 Encryption and decryption transformations

K denotes a set called the *key space*. An element of K is called a *key*.

Each element $e \in K$ uniquely determines a bijection from M to C , denoted by E_e is called an *encryption function* or an *encryption transformation*. E_e must be a bijection if the process is to be reversed and a unique plaintext message recovered for each distinct ciphertext.

For each $d \in K$, D_d denotes a bijection from C to M then D_d is called a *decryption function* or *decryption transformation*.

The process of applying the transformation E_e to a message $m \in M$ is usually referred to as *encrypting m* or the *encryption of m* .

The process of applying the transformation D_d to a ciphertext c is usually referred to as *decrypting c* or the *decryption* of c .

An *encryption scheme* consists of a set

$\{ E_e: e \in K \}$ of encryption transformations and a corresponding set

$\{ D_d: d \in K \}$ of decryption transformations with the property that for

each $e \in K$ there is a unique key $d \in K$ such that $D_d = E_e^{-1}$

that is, $D_d(E_e(m)) = m$ for all $m \in M$.

An encryption scheme is sometimes referred to as a *cipher*.

The keys e and d in the preceding definition are referred to as a *key pair* and sometimes denoted by (e, d) , e and d can be same also.

To *construct* an encryption scheme requires one to select a message space M , a ciphertext space C , a key space K , a set of encryption transformations $\{ E_e: e \in K \}$, and a corresponding set of decryption transformations $\{ D_d: d \in K \}$.

An encryption scheme may be used as follows for the purpose of achieving confidentiality. Two parties X and Y first secretly choose or secretly exchange a key pair (e, d) . At a subsequent point in time, if

X wishes to send a message $m \in M$ to Y, X computes $c = E_e(m)$ and transmits this to Y. Upon receiving c , Y computes $D_d(c) = M$ and hence recovers the original message M .

Having transformations which are very similar but characterized by keys means that if some particular encryption/decryption transformation is revealed then one does not have to redesign the entire scheme but simply change the key. It is a sound cryptographic practice to change the keys (encryption / decryption transformation) frequently.

A fundamental premise in cryptography is that the sets

$M, C, K, \{E_e : e \in K\}, \{D_d : d \in K\}$ are public knowledge.

When two parties wish to communicate securely using an encryption scheme, the only thing that they keep secret is the particular key pair (e, d) which they plan to use, and which they must decide in advance. One can gain additional security by keeping the class of encryption and decryption transformations secret but one should not base the security of the entire scheme on this approach. History has shown that maintaining the secrecy of the transformations is very difficult indeed.

An encryption scheme is said to be *breakable* if a third party, without prior knowledge of the key pair (e, d) , can systematically recover plaintext from corresponding cipher text within some appropriate time frame. An appropriate time frame will be a function of the useful

lifespan of the data being protected.

Recovering plain text without knowing the actual key is called crypt analysis. An encryption scheme can be broken by trying all possible keys to find out the actual key used by the communicating parties (assuming that the class of encryption functions is public knowledge). This is called an *exhaustive search* of the key space. It follows then that the possible number of keys (i.e., the size of the key space) should be large enough to make this approach computationally infeasible. It is the objective of designer of an encryption scheme to make sure that Exhaustive Key Search method will not help crypt analysis.

3.6 Symmetric Key Encryption

Symmetric-key encryption has a very long history, as recorded by Kahn [KAH 1967]. [DEN 1983] is a good source for many of the more well known schemes such as the Caesar cipher, Vigenfere and Beaufort ciphers, rotor machines (Enigma and Hagelin), running key ciphers. Also Davies and Price [DAV 1989] and Konheim [KON 1981] give description about many schemes

Beker and Piper [BEK 1982] give an indepth treatment, including cryptanalysis of several of the classical systems used in World War II.

Shannon's paper [SHA 1949] is considered the seminal work on secure communications. It is also an excellent source for descriptions of various well-known historical symmetric-key ciphers.

Hill ciphers [HIL 1929], is a class of substitution ciphers which substitute blocks using matrix methods. The idea of confusion and diffusion was introduced by Shannon [SHA 1949].

3.7 Public Key Cryptography

The main concept of public-key cryptography is that users can communicate securely - with privacy from eavesdroppers and assurance that messages exchanged are authentic - without first sharing secret information. The most notable microprocessor - related impact of public-key technology is perhaps in the area of integrated circuit cards, the development of which coincided with the maturation of public-key cryptography.

Internet users are prone to the so called *highwaymen*, called crackers, ranging from malicious pranksters to hardened terrorists. The Internet infrastructures are designed consisting of system mechanisms and protocols to prevent breach of security. This usually involves an intrusion detection system and data encryption on telecommunication services like the electronic mail. Security systems that detect deviations in a user's behavior can indicate only that a user may be an attacker, not what weak points were exploited to violate the security policy. So whether the Internet becomes secure depends entirely on the vendors that sell them and the users themselves.

Scrambling is a common approach used by conditional access systems to prevent unauthorized access to audio/visual data. The descrambling

keys are securely distributed to the receivers in the same transmission channel. Their protection is an important part of the key management problem. Although public-key cryptography provides a viable solution, alternative methods are sought for economy and efficiency. [ESK 2001] presents a key transport protocol based on secret sharing. It eliminates the need for a cipher, yet combines the advantages of symmetric and public-key ciphers.

One-way and trapdoor one-way functions are the basis for public-key cryptography. 1976 marked a major turning point in the history of cryptography. In several papers published in that year, Diffie and Hellman introduced the idea of public-key cryptography and gave concrete examples of how such a scheme might be realized.

The first paper on public-key cryptography was *Multiuser Cryptographic Techniques* by Diffie and Hellman [HEL 1976], presented at the National Computer Conference in June of 1976. Although the authors were not satisfied with the examples they cited, the concept was made clear.

In their landmark paper, Diffie and Hellman [DIF 1976] provided a more comprehensive account of public-key cryptography and described the first viable method to realize this elegant concept.

Another good source for the early history and development of the subject is Diffie [DIF 1992]. Nechvatal [NEC 1992] also provides a broad survey of public-key cryptography.

Merkle [MER 1978, MER 1979] independently discovered public-key cryptography, illustrating how this concept could be realized by giving an elegant and ingenious example now commonly referred to as the *Merkle puzzle scheme*.

In 1978 Rivest, Shamir, and Adleman [RIV 1978] discovered the first practical public key encryption and signature scheme, now referred to as RSA. The RSA scheme is based on another hard mathematical problem, the intractability of factoring large integers. This application of a hard mathematical problem to cryptography revitalized efforts to find more efficient methods to factor.

The 1980s saw major advances in this area but none which rendered the RSA system insecure. Another class of powerful and practical public key schemes was found by ElGamal in 1985. These are also based on the discrete logarithm problem.

One of the most significant contributions provided by public-key cryptography is the digital signature. In 1991 the first international standard for digital signatures (ISO/IEC 9796) was adopted. It is based on the RSA public key scheme.

In 1994 the U.S. Government adopted the Digital Signature Standard, a mechanism based on the Elgamal [ELG 1985] public key scheme.

The search for new public key schemes, improvements to existing cryptographic mechanisms, and proofs of security continues at a rapid pace. Various standards and infrastructures involving cryptography are

being put in place. Security products are being developed to address the security needs of an information intensive society.

The encryption method is said to be a *public key encryption scheme* if for each associated encryption/decryption pair (e, d), one key e (*the public key*) is made publicly available, while the other d (*the private key*) is kept secret. For the scheme to be *secure*, it must be computationally infeasible to compute d from e . To avoid ambiguity, a common convention is to use the term *private key* in association with public key cryptosystems, and *secret key* in association with symmetric key cryptosystems. This may be motivated by the line of thought - it takes two or more parties to *share* a secret, but a key is truly *private* only when one party alone knows it.

The internet public key infrastructure provides the secure digital certification required to establish a network of trust for public commerce. [BEN 2001] explores the details of the infrastructure.

Symmetric Key vs. Public Key Cryptography

Symmetric key and public key encryption schemes have various advantages and disadvantages, some of which are common to both. This section highlights a number of these and summarizes features pointed out in previous sections.

3.8.1 Advantages of Symmetric Key Cryptography

1. Symmetric-key ciphers can be designed to have high rates of data throughput. Some hardware implementations achieve encrypt rates of hundreds of megabytes per second, while software implementations may attain throughput rates in the megabytes per second range.
2. Keys for symmetric-key ciphers are relatively short.
3. Symmetric-key ciphers can be employed as primitives to construct various cryptographic mechanisms including pseudorandom number generators, hash functions and computationally efficient digital signature schemes.
4. Symmetric-key ciphers can be composed to produce stronger ciphers. Simple transformations which are easy to analyze, but on their own weak, can be used to construct strong product ciphers.
5. Symmetric-key encryption is perceived to have an extensive history, although it must be acknowledged that the invention of rotor machines earlier, much of the knowledge in this area has been acquired subsequent to the invention of the digital computer, and, in particular, the design of the Data Encryption Standard in the early 1970s.

3.8.2 Disadvantages of Symmetric Key Cryptography

1. In a two party communication, the key must remain secret at both ends.
2. In a large network, there are many key pairs to be managed. Consequently, effective key management requires the use of an unconditionally trusted TTP
3. In a two-party communication between entities A and B , sound cryptographic practice dictates that the key be changed frequently, and perhaps for each communication session.
4. Digital signature mechanisms arising from symmetric-key encryption typically require either large keys for the public verification function or the use of a TTP.

3.8.3 Advantages of Public Key Cryptography

1. Only the private key must be kept secret (authenticity of public keys must, however, be guaranteed).
2. The administration of keys on a network requires the presence of only a functionally trusted TTP as opposed to an unconditionally trusted TTP.

- 3 Depending on the mode of usage, a private key/public key pair may remain unchanged for considerable periods of time, e.g., many sessions (even several years).
- 4 Many public-key schemes yield relatively efficient digital signature mechanisms. The key used to describe the public verification function is typically much smaller than for the symmetric-key counterpart.
5. In a large network, the number of keys necessary may be considerably smaller than in the symmetric-key scenario.

3.8.4 Disadvantages of Public Key Encryption

1. Throughput rates for the most popular public-key encryption methods are several orders of magnitude slower than the best known symmetric key schemes.
2. Key sizes are typically much larger than those required for symmetric key encryption, and the size of public-key signatures is larger than that of tags providing data origin authentication from symmetric key techniques.
3. No public key scheme has been proven to be secure (the same can be said for block ciphers). The most effective public-key encryption schemes found to date have their security based on the presumed difficulty of a small set of number-theoretic problems.

4. Public key cryptography does not have as extensive a history as symmetric-key encryption, being discovered only in the mid 1970s.

Symmetric key and public key encryption have a number of complementary advantages. Current cryptographic systems exploit the strengths of each.

Public key encryption techniques may be used to establish a key for a symmetric-key system being used by communicating entities X and Y . In this scenario X and Y can take advantage of the long term nature of the public/private keys of the public-key scheme and the performance efficiencies of the symmetric-key scheme. The important points in practice are:

1. public key cryptography facilitates efficient signatures (particularly non-repudiation) and key management
2. symmetric key cryptography is efficient for encryption and some data integrity applications.
3. Private keys in public key systems must be larger (e.g., 1024 bits for RSA) than secret keys in symmetric key systems (e.g., 64 or 128 bits).
4. The most efficient attack on symmetric key systems is an

exhaustive key search and all known public key systems are subject to *short cut* attacks which are more efficient than exhaustive search. Consequently, for equivalent security, symmetric keys have bit lengths considerably smaller than that of private keys in public key systems, e.g., by a factor of 10 or more.

3.9 Block Ciphers

A *block cipher* is an encryption scheme which breaks up the plaintext messages to be transmitted into strings, called *blocks*, of a fixed length and encrypts one block at a time.

Most well-known symmetric-key encryption techniques are block ciphers. Two important classes of block ciphers are *substitution ciphers* and *transposition ciphers*. Product ciphers combine these.

Symmetric key block ciphers are the most prominent and important element in many cryptographic systems. Individually, they provide confidentiality. As a fundamental building block, their versatility allows construction of pseudorandom number generators, stream ciphers, and hash functions.

They may furthermore serve as a central component in message authentication techniques, data integrity mechanisms, entity authentication protocols, and (symmetric key) digital signature schemes.

No block cipher is ideally suited for all applications, even one offering a high level of security. This is a result of inevitable tradeoffs required in practical applications, including those arising from, for example, speed requirements and memory limitations (e.g., code size, data size, cache memory), constraints imposed by implementation platforms (e.g., hardware, software, chipboards), and differing tolerances of applications to properties of various modes of operation. In addition, efficiency must typically be traded off against security.

Of the many block ciphers currently available, high profile and/or well-studied algorithms of greatest practical interest are considered. Among these, DES is paramount; FEAL has received both serious commercial backing and a large amount of independent cryptographic analysis; and IDEA (originally proposed as a DES replacement) is widely known and highly regarded. Other recently proposed ciphers of both high promise and high profile are SAFER and RC5.

The extensive and particularly reliable survey by Diffie and Hellman [DIE 1979], providing a broad introduction to cryptography is noteworthy for its treatment of Hagelin and rotor machines.

Aside from the appearance of DES [FIP 1977] in the mid 1970s and FEAL [MIY 1991] later in the 1980s, prior to 1990 only a few fully-specified serious symmetric block cipher proposals were widely available or discussed.

With the increasing feasibility of exhaustive search on 56-bit DES keys, the period 1990-1995 resulted in a large number of proposals,

beginning with PES [LAI 1991], the preliminary version of IDEA [LAI 1991 A].

Lai [LAI 1992] provides an excellent concise introduction to block ciphers, including a lucid discussion of design principles.

Rivest and Sherman [RIV 1983] provide a unified framework for randomized encryption. Use of random *salt* appended to passwords prior to password encryption in some operating systems is explained in detail in this.

The four basic modes of operation (including CFB bit OFB feedback) were originally defined specifically for DES in 1980 by FIPS 81 [FIP 1981] and in 1983 by ANSI X3.106 [ANS 1983], while ISO 8732 [ISO 1988] and ISO/EC 10116 [ISO 1991], respectively, defined these modes for general 64-bit and general n - bit block ciphers, mandating n- bit OFB feedback.

Brassard [BRA 1988] gives a concise summary of modes of operation while Davies and Price [DAV 1983] provide a comprehensive discussion, including OFB cycling.

OFB cycling is explained in detail in Jueneman [JUE 1983] and Davies and Parkin [DAV 1983] also.

A method for encrypting incomplete CBC final blocks without data expansion, which is important if plaintext must be encrypted and returned into its original store is explained in Voydock and Kent

[VOY 1985]. ISO/IEC 10116 [ISO 1991] specifies the CFB variations and provides extensive discussion of properties of the various modes.

3.9.1 Classical Ciphers and Historical Development

Kahn [KAH 1967] is the definitive historical reference for classical ciphers and machines up to 1967. The selection of classical ciphers presented largely follows Shannon's lucid 1949 paper [SHA 1949].

Polyalphabetic ciphers were invented in 1467 by the Florentine architect Alberti, who devised a cipher disk with a larger outer and smaller inner wheel, respectively indexed by plaintext and ciphertext characters.

The Playfair cipher was developed by the British scientist, Wheatstone and it was popularized by Playfair in England. It was used as a British field cipher [KAH 1967].

The Jefferson cylinder was designed by American statesman, Jefferson in 1817. In 1867, Wheatstone displayed an independently developed device called the *Wheatstone disc*, receiving greater attention although less secure (having disks of respectively 26 and 27 characters, the extra character a plaintext space).

Vernam Cipher, [VER 1926], was developed for use of telegraph encryption in 1917. Vernam's device combined a stream of plaintext (5-bit Baudot coded) characters, via XOR, with a key stream of 5-bit (key) values. Though Vernam cipher involves only 32 alphabets,

provides more security than rotor machines with a far greater number of alphabets.

The matrix cipher was proposed in 1929 by Hill [HIL 1929], providing a practical method for polygraphic substitution, albeit a linear transformation susceptible to known plaintext attack. Recent contributions on homophonic substitution include Gunther [GUN 1988] and Jendal, Kuhn, and Massey [JEN 1990].

Shannon [SHA 1951] and Cover and King [COV 1978] are regarding redundancy. Unicity distance was defined by Shannon [SHA 1949]. Related issues are discussed in detail in various appendices of Meyer and Matyas [MEY 1982]. Random cipher model is due to Shannon [SHA 1949] and Hellman [HEL 1977].

Diffie and Hellman [DIF 1979] give an instructive overview of rotor machines, (also Denning [DEN 1983]) and note their use in World War II by the Americans in their highest level systems.

Beker and Piper [BEK 1982] provide technical details of the Hagelin M-209, as does Kahn [KAH 1967].

Davies and Price [DAV 1989] briefly discuss the Enigma, the encryption method used by Germans in World War II.

The Japanese PURPLE cipher, used during World War II, was a poly alphabetic cipher cryptanalysed in August 1940 [KAH 1967] by Friedman's team in the U.S. Signal Intelligence Service. The earlier

RED cipher used two rotor arrays; preceding it, the ORANGE system implemented a vowels-to-vowels, consonants to-consonants cipher using sets of rotors.

Shannon [SHA 1949] explored the idea of the product of two ciphers, noted the principles of confusion and diffusion, and introduced the idea of a *mixing transformation* (suggesting a preliminary transposition followed by a sequence of alternate substitution and simple linear operations), and combining ciphers in a product using an intervening transformation. Transposition and substitution, respectively, rest on the principles of diffusion and confusion. Harpes, Kramer, and Massey [HAR 1985] discuss a general model for iterated block ciphers.

The name *Lucifer* is associated with two very different algorithms. The first employs (bitwise nonlinear) 4 x 4 invertible S-boxes; the second, closely related to DES, is described by Smith [SMI 1971] and also Sorkin [SOR 1984]. Principles related to both are discussed by Feistel, Notz, and Smith [FEI 1988]. Both are analyzed by Biham and Shamir [BIH 1993].

3.9.2 Data Encryption Standard (DES)

DES resulted from IBM's submission to the 1974 U.S. National Bureau of Standards (NBS) solicitation for encryption algorithms for the protection of computer data. The original specification is the 1977 U.S. Federal Information Processing Standards Publication 46 [FIP 1977], reprinted in its entirety as Appendix A in Meyer and Matyas [MEY 1982].

DES is now specified in FIPS 46-2, which succeeded FIPS 46-1; the same cipher is defined in the American standard ANSI X3.92 [ANS 1981] and referred to as the Data Encryption Algorithm (DEA).

Differences between FIPS 46/46-1 and ANSI X3.92 included the following: these earlier FIPS required that DES be implemented in hardware and that the parity bits be used for parity; ANSI X3.92 specifies that the parity bits may be used for parity. Although no purpose was stated by the DES designers for the permutations IP and IP^{-1} , Preneel [PRE 1994] provided some evidence of their cryptographic value in the CFB mode.

FIPS 81 [FIP 1981] specifies the common modes of operation. Davies and Price [DAV 1989] provide a comprehensive discussion of both DES and modes of operation are given in detail in Diffie and Hellman [DIF 1979], and the extensive treatment of Meyer and Matyas [MEY 1982]. The survey of Smid and Branstad [SMI 1992] discusses DES, its history, and its use in the U.S. government. Test vectors for various modes of DES, including the ECB vectors, can be found in ANSI X3.106 [ANS 1983].

The 1981 publication FIPS 74 [FIP 1981] notes that DES is not (generally) commutative under two keys, and summarizes weak and semi-weak keys using the term *dual keys* to include both (weak keys being self-dual). Moore and Simmons [MOO 1987] pursue weak and semi-weak DES keys and related phenomena more rigorously.

The 56-bit keylength of DES was criticized from the outset as being too small (e.g. Diffie and Hellman [DIF 1977]). Claims which have repeatedly arisen and been denied (e.g. Tuchman [TUC 1979]) over the past 25 years regarding built-in weaknesses of DES (e.g., trap-door S-boxes) remain un-substantiated. It is significant that if the permutation group were closed under composition, DES would fall to a known-plaintext attack requiring 2^{28} steps. Kaliski, Rivest, and Sherman [KAL 1988], whose cycling experiments provided strong evidence against this. Campbell and Wiener [CAM 1993] prove the fact conclusively (and give the stated lower bound), through their own cycling experiments utilizing collision key search and an idea outlined earlier by Coppersmith [COP 1986] for establishing a lower bound on the group size: they attribute to Coppersmith the same result, which may also be deduced from the cycle lengths published by Moore and Simmons [MOR 1986].

Countless papers have analyzed various properties of DES. Subsequent to the discovery of differential cryptanalysis (DC) by Biham and Shamir, Coppersmith [COP 1994] explains how DES was specifically designed 15 years earlier to counter DC, citing national security concerns regarding the design team publishing neither the attack nor design criteria; then gives the (relevant) design criteria - some already noted by others. Hellman [HEL 1976] describes DES S-boxes and the permutation P , explaining how these preclude DC.

DES was not specifically designed to preclude linear cryptanalysis (LC). Matsui [MAT 1995] suggests that DES can be strengthened against DC and LC by carefully re-arranging the order of 8 S boxes.

DES key has actually been recovered by Matsui [MAT 1993] using LC under experimental conditions (using 2^{43} known-plaintext pairs from randomly generated plaintexts, and 2^{43} complexity running twelve 99 MHz machines over 50 days).

Ben-Aroya and Biham [BEN 1996] note that often suggestions to redesign DES, some based on design criteria and attempts to specifically resist DC, have resulted in weaker systems, including the RDES (randomized DES) proposal of Koyama and Terada [KOY 1993], which fall prey to variant attacks. The lesson is that in isolation, individual design principles do not guarantee security.

DES alternatives are sought not only due to the desire for a key length exceeding 56 bits, but also because its bit-oriented operations are inconvenient in conventional software implementations, often resulting in poor performance: this makes triple-DES less attractive.

3.9.2 Fast Data Encipherment Algorithm (FEAL)

FEAL stimulated the development of a sequence of advanced cryptanalytic techniques of unparalleled richness and utility. While it appears to remain relatively secure when iterated a sufficient number of rounds (e.g., 24 or more), this defeats its original objective of speed. FEAL-4 as presented at *Eurocrypt 87* was found to have certain vulnerabilities by Den Boer, resulting in increasing FEAL to 8 rounds in the final proceedings - Shimizu and Miyaguchi [SHI 1988], Miyaguchi, Shiraishi, and Shimizu [MIY 1988]. In 1990, Gilbert and Chasse [GIL 1991] devised a chosen-plaintext attack (called a

statistical meet-in-the-middle attack) on FEAL-8 requiring 10000 pairs of plaintexts, the bitwise XOR of each pair being selected to be an appropriate constant (thus another early variant of differential cryptanalysis).

FEAL- N with N rounds, and its extension FEAL-NX with 128-bit key were then published by Miyaguchi [MIY 1991, MIY 1990] who nonetheless opined that chosen-plaintext attacks on FEAL-8 were not practical threats. However, improved chosen-plaintext attacks were subsequently devised, as well as known-plaintext attacks.

A statistical method of Tardy-Corffdir and Gilbert [TAR 1992] allowed a known-plaintext attack on FEAL - 4 (1000 texts; or 200 in an announced improvement) and FEAL - 6 (20000 texts), involving linear approximation of FEAL S-boxes.

Thereafter, the first version of linear cryptanalysis (LC) introduced by Matsui and Yamagishi [MAT 1993] allowed known-plaintext attack of FEAL-4 (5 texts, 6 minutes on a 25MHz 68040 processor). FEAL-6 (100 texts, 40 minutes), and FEAL-8 (2^{28} texts, in time equivalent to exhaustive search on 50-bit keys); the latter betters the 2^{38} texts required for FEAL-8 by Biham and Shamir [BIH 1991] in their known-plaintext conversion of differentia cryptanalysis (DC).

Biham and Shamir [BIH 1993] later implemented a DC chosen plaintext attack recovering FEAL-8 keys in two minutes on a PC using 128 chosen pairs, the program requiring 280 K bytes of storage.

Biham [BIH 1995] subsequently used LC to defeat FEAL-8 with 2^{24} known-plaintexts in 10 minutes on a personal computer. Ohta and Aoki [OHT 1994] suggest that FEAL-32 is as secure as DBS against DC, while FEAL-16 is as secure as DES against certain restricted forms of LC.

Differential Linear Cryptanalysis was introduced by Langford and Hellman [LAN 1994], combining linear and differential cryptanalysis to allow a reduced 8-round version of DES to be attacked with fewer chosen-plaintexts than previous attacks. Aoki and Ohta [AOK 1996] refined these ideas for FEAL-8 yielding a differential-linear attack requiring only 12 chosen texts and 35 days of computer time.

9.4 International Data Encryption Algorithm (IDEA)

The primary reference for IDEA is Lai [LAI 1992]. A preliminary version introduced by Lai and Massey [LAI 1991] was named PES (Proposed Encryption Standard).

Lai, Massey, and Murphy [LAI 1991 A] showed that a generalization of differential cryptanalysis (DC) allowed recovery of PES keys, albeit requiring all 2^{64} possible ciphertexts

Daemon [DAE 1994, DAE 1995] identifies several classes of so-called *weak keys for IDEA*, and notes a small modification to the key schedule to eliminate them. The largest is a class of 2^{51} keys for which membership can be tested in two encryptions plus a small number of computations, where after the key itself can be recovered using 16

chosen plaintext-difference encryptions, of the order of 2^{16} group operations, plus 2^{17} key search encryptions. A smaller number of weak key blocks were observed earlier by Lai [LAI 1992], and dismissed as inconsequential.

The analysis of Meier [MEI 1994] revealed no attacks feasible against full 8-round IDEA, and supports the conclusion of Lai [LAI 1992] that IDEA appears to be secure against DC after 4 of its 8 rounds.

Daemon [DAE 1995] also references attacks on reduced-round variants of IDEA. While linear cryptanalysis (LC) can be applied to any iterated block cipher, Harpes, Kramer, and Massey [HAR 1995] provide a generalization thereof; IDEA and SAFER K-64 are argued to be secure against this particular generalization.

3.9.5 Secure And Fast Encryption Routine (SAFER)

Massey [MEI 1994] introduced SAFER K-64 with a 64-bit key and initially recommended 6 rounds, giving a reference implementation and test vectors. Massey [MAS 1995] then published SAFER K-128, differing only in its use of a non-proprietary key schedule accommodating 128-bit keys. Massey [MAS 1995] gave further justification for design components of SAFER K-64.

Vaudenay [VAU 1995] showed that SAFER K-64 is weakened if the S-box mapping is replaced by a random permutation.

Knudsen [KNU 1995] proposed the modified key schedule after finding a weakness in 6-round SAFER K-64 that, while not of practical concern for encryption (with 2^{45} chosen plaintexts, it finds 8 bits of the key), permitted collisions when using the cipher for hashing. This and a subsequent certification attack on SAFER K-64 by S. Murphy led Massey to advise adoption of the new key schedule, with the resulting algorithm distinguished as SAFER SK-64 with 8 rounds recommended (minimum 6, maximum 10); an analogous change to the 128-bit key schedule yields SAFER SK-128 for which 10 rounds remain recommended (maximum 12).

A new variant of Differential Cryptanalysis by Knudsen and Berson [KNU 1996] using *truncated differentials* yields a certification attack on 5-round SAFER K-64 with 245 chosen plaintexts, the attack, which does not extend to 6 rounds, indicates that security is less than argued by Massey [MAS 1995], who also notes that preliminary attempts at linear cryptanalysis of SAFER were unsuccessful.

19.6 RC 5

RC5 was designed by Rivest [RIV 1995], and published along with a reference implementation. The magic constants are based on the golden ratio and the base of natural logarithms. The data-dependent rotations (which vary across rounds) distinguish RC5 from iterated ciphers which have identical operations each round.

A preliminary examination by Kaliski and Yin [KAL 1995] suggested that, while variations remain to be explored, standard linear and

differential cryptanalysis appear impractical for RC5-32 (64-bit blocksize) for $T = 12$: their differential attacks on 9 and 12 round RC5 require, respectively, 2^{45} , 2^{62} chosen-plaintext pairs, while their linear attacks on 4,5, and 6-round RC5-32 require, respectively, 2^{37} , 2^{47} , 2^{57} known plaintexts. Both attacks depend on the number of rounds and the blocksize, but not the byte-length of the input key (since sub keys are recovered directly).

Knudsen and Meier [KNU 1996] subsequently presented differential attacks on RC5 which improved on those of Kaliski and Yin by a factor up to 128, and showed that RC5 has so called *weak keys* (independent of the key schedule) for which these differential attacks perform even better.

9.7 Other Block Ciphers

LOKI 91 was proposed as a DES alternative with a larger 64 bit key, a matching 64 bit blocksize and 16 rounds. It differs from DES mainly in key scheduling and the f function.

LOKI 89 was introduced by Brown, Pieprzyk, and Seberry [BRO 1990] and renamed LOKI'89 after the discovery of weaknesses lead to the introduction of LOKI'91 by Brown et al. [BRO 1993]. Knudsen [KNU 1993] noted each LOKI'89 key fell into a class of 16 equivalent keys, and the differential cryptanalysis of Biham and Shamir [BIH 1992] was shown to be effective against reduced-round versions. LOKI 91 failed to succumb to differential analysis by Knudsen [KNU 1993]; Tokita et al, [TOK 1995] later confirmed the optimality of Knudsen's

characteristics, suggesting that LOKI'89 and LOKI'91 were resistant to both ordinary linear and differential cryptanalysis. However, neither should be used for hashing as originally proposed by Knudsen [KNU 1993] or in other modes described in Preneel [PRE 1993]. Moreover, both are susceptible to *related-key attacks* popularized by Biham [BIH 1994, BIH 1994A]. Distinct from these are *key clustering attacks* of Diffie and Hellman [DIF 1979], wherein a cryptanalyst first finds a key *close* to the correct key, and then searches a cluster of "nearby" keys to find the correct one.

CAST is a design procedure for a family of DES-like ciphers, featuring fixed $m \times 71$ bit S-boxes based on bent functions. Adams and Tavares [ADA 1993] examine the construction of large S-boxes resistant to differential cryptanalysis and give a partial example (with 64-bit block length and 8×32 bit S-boxes) of a CAST cipher. CAST ciphers have variable keysize and numbers of rounds. Rijmen and Preneel [RIJ 1995] presented a cryptanalytic technique applicable to Feistel ciphers.

BLOWFISH is a 16-round DES-like cipher due to Schneier [SCH 1994], with 64-bit blocks and keys of length up to 448 bits. The computationally intensive key expansion phase creates eighteen 32-bit subkeys plus four 8×32 bit S-boxes derived from the input key, for a total of 4168 bytes. Preliminary analysis of Blowfish is given in Vaudenay [VAU 1996].

3-WAY is a block cipher with 96-bit blocksize and keysize, due to Daemen [DAE 1995] and introduced by Daemen, Govaerts, and Vandewalle [DAE 1994] along with a reference C implementation and

test vectors. It was designed for speed in both hardware and software, and to resist differential and linear attacks. Its core is a 3-bit nonlinear S-box and a linear mapping representable as polynomial multiplication.

SHARK is an SP-network block cipher due to Rijmen et al. [RIJ 1996] which may be viewed as a generalization of SAFER, employing highly nonlinear S-boxes and the idea of MDS codes for diffusion to allow a small number of rounds to suffice.

BEAR and LION of Anderson and Biham [AND 1996] are 3-round unbalanced Feistel networks, motivated by the earlier construction of Luby and Rackoff [LUB 1988] which provides a provably secure (under suitable assumptions) block cipher from pseudorandom functions using a 3-round Feistel structure. SHARK, BEAR, and LION all remain to be subjected to independent analysis in order to substantiate their conjectured security levels.

SKIPJACK is a classified block cipher whose specification is maintained by the U.S. National Security Agency (NSA). FIPS 185 [FIP 1994] notes that its specification is available to organizations entering into a Memorandum of Agreement with the NSA, and includes interface details (e.g., it has an 80-bit secret key). Roe [ROE 1995] gives details regarding curious results on the cyclic closure tests on SKIPJACK and evidence related to the size of the cipher keyspace.

COST 28147-89 is a Soviet government encryption algorithm with a 32-round Feistel structure and unspecified S-boxes, Charnes et al. [CHA 1995].

WAKE is a block cipher due to Wheeler [WHE 1994] employing a key-dependent table, intended for fast encryption of bulk data on processors with 32-bit words.

TEA (Tiny Encryption Algorithm) is a block cipher proposed by Wheeler and Needham [WHE 1995].

3.10 Stream Ciphers

Stream ciphers are an important class of encryption algorithms. They encrypt individual characters (usually binary digits) of a plaintext message one at a time, using an encryption transformation which varies with time. By contrast, *block ciphers* tend to simultaneously encrypt groups of characters of a plaintext message using a fixed encryption transformation. Stream ciphers are generally faster than block ciphers in hardware, and have less complex hardware circuitry. They are also more appropriate, and in some cases mandatory (e.g., in some telecommunications applications), when buffering is limited or when characters must be individually processed as they are received. Because they have limited or no error propagation, stream ciphers may also be advantageous in situations where transmission errors are highly probable.

There is a vast body of theoretical knowledge on stream ciphers, and various design principles for stream ciphers have been proposed and extensively analyzed. However, there are relatively few fully-specified stream cipher algorithms in the open literature. This unfortunate state is

due to the fact that most stream ciphers used in practice tend to be proprietary and confidential. By contrast, numerous concrete block cipher proposals have been published, some of which have been standardized or placed in the public domain. Nevertheless, because of their significant advantages, stream ciphers are widely used today, and one can expect increasingly more concrete proposals in the coming years.

Stream ciphers can be either symmetric-key or public-key. The focus of this review is on symmetric-key stream ciphers. Blum-Goldwasser probabilistic public-key encryption scheme is an example of a public-key stream cipher.

Block ciphers process plaintext in relatively large block (e.g., $n \geq 64$ bits). The same function is used to encrypt successive blocks, thus pure block ciphers are *memoryless*. In contrast, stream ciphers process plaintext in blocks as small as a single bit, and the encryption function may vary as plaintext is processed; thus stream ciphers are said to have memory. They are sometimes called *state ciphers* since encryption depends on not only the key and plaintext, but also on the current state. This distinction between block and stream ciphers is not definitive adding a small amount of memory to a block cipher results in a stream cipher with large blocks.

Stream ciphers are commonly classified as being *synchronous* or *self-synchronizing*.

A *synchronous* stream cipher is one in which the keystream is generated independently of the plaintext message and of the ciphertext.

In a synchronous stream cipher, both the sender and receiver must be *synchronized* - using the same key and operating at the same position (state) within that key - to allow for proper decryption. If synchronization is lost due to ciphertext digits being inserted or deleted during transmission, then decryption fails and can only be restored through additional techniques for resynchronization. Techniques for resynchronization include re-initialization, placing special markers at regular intervals in the ciphertext, or, if the plaintext contains enough redundancy, trying all possible keystream offsets.

A ciphertext digit that is modified (but not deleted) during transmission does not affect the decryption of other ciphertext digits. Insertion, deletion, or replay of ciphertext digits by an active adversary causes immediate loss of synchronization, and hence might possibly be detected by the decryptor. An active adversary might possibly be able to make changes to selected ciphertext digits, and know exactly what affect these changes have on the plaintext. Additional mechanisms must be employed in order to provide data origin authentication and data integrity guarantees.

A *binary additive stream cipher* is a synchronous stream cipher in which the keystream, plaintext, and ciphertext digits are binary digits, and the output function is the XOR function.

A *self-synchronizing* or *asynchronous* stream cipher is one in which the key stream is generated as a function of the key and a fixed number of previous ciphertext digits.

The most common presently-used self-synchronizing stream ciphers are based on block ciphers in 1-bit cipher feedback mode. Self-synchronization is possible if cipher text digits are deleted or inserted, because the decryption mapping depends only on a fixed number of preceding ciphertext characters. Such ciphers are capable of re-establishing proper decryption automatically after loss of synchronization, with only a fixed number of plaintext characters unrecoverable.

The state of a self-synchronization stream cipher depends on t previous ciphertext digits. If a single ciphertext digit is modified (or even deleted or inserted) during transmission, then decryption of up to t subsequent ciphertext digits may be incorrect, after which correct decryption resumes.

Any modification of ciphertext digits by an active adversary causes several other ciphertext digits to be decrypted incorrectly, thereby improving (compared to synchronous stream ciphers) the likelihood of being detected by the decryptor. It is more difficult (than for synchronous stream ciphers) to detect insertion, deletion, or replay of ciphertext digits by an active adversary. This illustrates that additional mechanisms must be employed in order to provide data origin authentication and data integrity guarantees.

Since each plaintext digit influences the entire following ciphertext, the statistical properties of the plaintext are dispersed through the ciphertext. Hence, self-synchronizing stream ciphers may be more resistant than synchronous stream ciphers against attacks based on plaintext redundancy.

3.10.1 Stream ciphers based on LFSRs

Linear feedback shift registers (LFSR) are widely used in keystream generators because they are well-suited for hardware implementation, produce sequences having large periods and good statistical properties, and are readily analyzed using algebraic techniques.

Unfortunately, the output sequences of LFSRs are also easily predictable.

Since a well-designed system should be secure against known-plaintext attacks, an LFSR should never be used by itself as a keystream generator. Nevertheless, LFSRs are desirable because of their very low implementation costs.

3.11.2 Other Stream Ciphers

While the LFSR-based stream ciphers are well-suited to hardware implementation, they are not especially amenable to software implementation. This has led to several recent proposals for stream ciphers designed particularly for fast software implementation. Most of these proposals are either proprietary, or are relatively new and have

not received sufficient scrutiny from the cryptographic community. Two promising stream ciphers specifically designed for fast software implementation are SEAL and RC4.

SEAL (Software Optimized Encryption Algorithm) is a binary additive stream cipher that was proposed in 1993. It is one of the few stream ciphers that was specifically designed for efficient software implementation and, in particular, for 32-bit processors.

RC4 is used in commercial products and has a variable key-size. Other widely used stream ciphers not based on LFSRs are the Output Feedback (OFB) and Cipher Feedback (CFB) modes of block ciphers.

Rueppel [RUE 1986] provides a solid introduction to the analysis and design of stream ciphers, an updated and more comprehensive survey is given in Rueppel [RUE 1992].

A stream cipher based on a nonlinear mixing of the outputs of three linear feedback shift registers was proposed by Chan and Cheng. This cipher is vulnerable to a Meier-Steffelbach correlation attack and it uses the outputs of three linear feedback shift registers (LFSR). Once the states of the second and third LFSRs are known, it is easy to recover the state of the first. Since the entire state of the generator may be recovered from a portion of its known output, the stream cipher is insecure according to [BLA 1998].

One technique for solving the re-synchronization problem with synchronous stream ciphers is to have the receiver send a

resynchronization request to the sender, whereby a new internal state is computed as a (public) function of the original internal state (or key) and some public information (such as the time at the moment of the request). Daemon, Govaerts, and Vandewalle [DAE 1994] showed that this approach can result in a total loss of security for some published stream cipher proposals.

Proctor [PRO 1985] considered the trade-off between the security and error propagation problems that arise by varying the number of feedback cipher text digits. Maurer [MAU 1991] presented various design approaches for self-synchronizing stream ciphers that are potentially superior to designs based on block ciphers, both with respect to encryption speed and security.

The results on the expected linear complexity and linear complexity profile of random sequences are given in Chapter 4 of Rueppel [RUE 1986]. Dai and Yang [DAI 1991] had obtained bounds for the expected linear complexity of an n -periodic sequence for each possible value of n . The bounds imply that the expected linear complexity of a random periodic sequence is close to the period of the sequence.

There are numerous other algorithms for computing the linear complexity of a sequence. For example, Games and Chan [GAM 1983] and Robshaw [BOB 1994] present efficient algorithms for determining the linear complexity of binary sequences of period 2^n : these algorithms have limited practical use since they require an entire cycle of the sequence.

Jansen and Boeke [JAN 1990] defined the *maximum order complexity* of a sequence to be the length of the shortest (not necessarily linear) feedback shift register (FSR) that can generate the sequence. The expected maximum order complexity of a random binary sequence of length n is approximately $2 \lg n$.

Klapper and Goresky [KLA 1995] introduced a new type of feedback register called a *feedback with carry shift register* (FCSR), which is equipped with auxiliary memory for storing the (integer) carry. An FCSR is similar to an LFSR, except that the contents of the tapped stages of the shift register are added *as integers* to the current content of the memory to form a sum S . The least significant bit of S (i.e., $S \bmod 2$) is then fed back into the first (leftmost) stage of the shift register, while the remaining higher order bits are retained as the new value of the memory. If the FCSR has L stages, then the space required for the auxiliary memory is at most $\lg L$ bits.

Any periodic binary sequence can be generated by a FCSR. The *2-adic span* of a periodic sequence is the number of stages and memory bits in the smallest FCSR that generates the sequence. Let s be a periodic sequence having a 2-adic span of T ; note that T is no more than the period of s . Klapper and Goresky [KLA 1995] presented an efficient algorithm for finding an FCSR of length T which generates s .

Selection of connection polynomials were essentially first pointed out by Meier and Staffelbach [MEI 1988] and Chepyzhov and Smeets [CHE 1991] in relation to fast correlation attacks on regularly clocked LFSRs. Similar observations were made by Coppersmith, Krawczyk,

and Mansour [COP 1994] in connection with the shrinking generator. More generally, to withstand sophisticated correlation attacks, the connection polynomials should not have low-weight polynomial multiples whose degrees are not sufficiently large.

Klapper [KLA 1994] provides examples of binary sequences having high linear complexity, but whose linear complexity is low when considered as sequences over a larger finite field. This demonstrates that high linear complexity by itself is inadequate for security. It is proven by Rueppel and Staffelbach [RUE 1985].

The correlation attack on nonlinear combination generators was first developed by Siegenthaler [SIE 1985], and estimates were given for the length of the observed keystream required for the attack to succeed with high probability. The importance of correlation immunity to nonlinear combining functions was pointed out by Siegenthaler [SIE 1984], who showed the tradeoff between high correlation immunity and high nonlinear order. Meier and Staffelbach [MEI 1989] presented two new so-called *fast correlation attacks* which are more efficient than Siegenthaler's attack in the case where the component LFSRs have sparse feedback polynomials, or if they have low-weight polynomial multiples.

A comprehensive survey of correlation attacks on LFSR-based stream ciphers is the paper by Golic [GOL 1994]; the cases where the combining function is memoryless or with memory, as well as when the LFSRs are clocked regularly or irregularly, are all considered.

The summation generator was proposed by Rueppel [RUE 1986]. Meier and Staffelbach [MEI 1992] presented correlation attacks on combination generators having memory, cracked the summation generator having only two component LFSRs, and as a result recommended using several LFSRs of moderate lengths rather than just a few long LFSRs in the summation generator. Dawson [DAW 1993] presented another known-plaintext attack on summation generators having two component LFSRs, which requires fewer known keystream bits than Meier and Staffelbach's attack. Dawson's attack is only faster than that of Meier and Staffelbach in the case where both LFSRs are relatively short.

Recently, Klapper and Goresky [KLA 1995] showed that the summation generator *has* comparatively low 2-adic span.

Blocher and Dichtl [BLO 1994] proposed a fast software stream cipher called *FISH* (Fibonacci Shrinking generator), which is based on the shrinking generator principle applied to the lagged Fibonacci generator of Knuth [KNU 1994].

Anderson [AND 1995] presents a known-plaintext attack on FISH which requires a few thousand 32-bit words of known plaintext and a work factor of about 240 computations.

Wolfram [WOL 1986] proposed a stream cipher based on one-dimensional cellular automata with nonlinear feedback. Meier and Staffelbach [MEI 1991] presented a known-plain text attack on this cipher which demonstrated that key lengths of 127 bits suggested by

Wolfram [WOL 1991] are insecure; Meier and Staffelbach recommend key sizes of about 1000 bits.

3.11 Crypt Analysis

Standard references for classical cryptanalysis include Friedman [FRI 1944], Gaines [GAI 1956], and Sinkov [SIN 1968]. More recent books providing material on classical ciphers, machines, and cryptanalytic examples include Beker and Piper [BEK 1982], Meyer and Matyas [MEY 1982], Denning [DEN 1983], and Davies and Price [DAV 1989].

The most significant cryptanalytic advances over the 1990-1995 period were Matsui's linear cryptanalysis [MAT 1994, MAT 1994 A], and the differential cryptanalysis of Biham and Shamir [BIH 1993]. Extensions of these included the differential-linear analysis by Langford and Hellman [LAN 1994], and the truncated differential analysis of Knudsen [KNU 1995].

Basic theories on various linear cryptanalysis, methods are given in Biham [BIH 1994], Matsui and Yamagishi [MAT 1993].

Friedman teaches how to cryptanalyze running-key ciphers in his Riverbank Publication no. 16, *Methods for the Solution of Running-Key Ciphers*, the two basic techniques are outlined by Diffie and Hellman [DIF 1976].

Additional background on differential cryptanalysis is provided by many authors including Lai [LAI 1992], Lai, Massey, and Murphy [LAI 1991], and Coppersmith [COP 1994]; although more efficient 6-round attacks are known, Stinson [STI 1995] provides detailed examples of attacks on 3-round and 6-round DES. An elaborative description regarding both linear and differential cryptanalysis are available in Knudsen [KNU 1994] and Kaliski and Yin [KAL 1995].

Regarding text dictionary and matching ciphertext attacks a vivid description is given in Coppersmith, Johnson, and Matyas [COP 1996].

The 1977 exhaustive DES key search machine proposed by Diffie and Hellman [DIF 1977] contained 10 DES chips, with estimated cost US\$20 million (1977 technology) and 12-hour expected search time. Diffie and Hellman noted the feasibility of a ciphertext-only attack, and that attempting to preclude exhaustive search by changing DES keys more frequently, at best, doubles the expected search time before success.

Subsequently Wiener [WEI 1993] provided a gate-level design for a machine (1993 technology) using 57600 DES chips with expected success in 3.5 hours. Each chip contains 16 pipelined stages, each stage completing in one clock tick at 50 MHz; a chip with full pipeline completes a key test every 20 nanoseconds, providing a machine of 57600×50 times faster than the 1142 years noted in FIPS 74 [FIP 1981] as the time required to check 255 keys if one key can be tested each microsecond.

Comparable key search machines of equivalent cost by Eberle [EBE 1993] and Wayner [WAY 1993] are, respectively, 55 and 200 times slower, although the former does not require a chip design, and the latter uses a general-purpose machine. Wiener also noted adaptations of the ECB known-plaintext attack to other 64-bit modes (CBC, OFB, CFB) and 1-bit and 8-bit CFB.

Even and Goldreich [EVE 1985] discuss the unicity distance of cascade ciphers under known plaintext attack, present a generalized time-memory meet-in-the-middle trade off, and give several other concise results on cascades, including that under reasonable assumptions, the number of permutations realizable by a cascade of L random cipher stages is, with high probability.

Diffie and Hellman [DIF 1977] noted the meet-in-the-middle attack on double encryption, motivating their recommendation that multiple encipherment, if used, should be at least three-fold.

Hoffman [HOF 1977] supports the above argument suggesting E-E-E triple encryption with three independent keys. Merkle's June 1979 thesis [MER 1979] explains the attack on two-key triple-encryption.

Another paper on the same topic is Merkle and Hellman [MER 1981]. Tuchman's proposal of two-key E-D-E triple encryption is given in [TUC 1979]. It recommends that E-D-E be used with three independent keys.

Coppersmith, Johnson, and Matyas [COP 1996] propose construction for a triple-DES algorithm.

Other techniques intended to extend the strength of DES include the *DESX* proposal of Rivest as analyzed by Kilian and Rogaway [KIL 1996], and the work of Biham and Biryukov [BIH 1995].

Hellman [HEL 1980] proposes a time-memory trade off for exhaustive key search on an n -bit cipher requiring a chosen-plaintext attack.

Denning [DEN 1983] suggests that search time can be reduced somewhat by use of Rivest's suggestion of distinguished points. Kusuda and Matsumoto [KUS 1996] recently extended this analysis.

Fiat and Naor [FIA 1991] pursue time-memory tradeoffs for more general functions. Amirazizi and Hellman [AMI 1988] note that time-memory tradeoff with constant time memory product offers no cost advantage over exhaustive search. Using standard parallelization techniques, they propose a search machine architecture for which doubling the machine budget (cost) increases the solution rate four-fold. This approach can be applied to exhaustive key search on double-encryption, as well as the parallel collision search technique of Oorschot and Wiener [OOR 1991, OOR 1994].

Biham's [BIH 1995] analysis on DES and FEAL shows that, in many cases, the use of intermediate data as feedback into an intermediate stage reduces security.

Even and Goldreich [EVE 1985] prove that a cascade is as strong as any of its component ciphers as an adversary can not exploit statistics of the underlying plaintext.

3.12 Digital signatures

The concept of a digital signature was introduced by Diffie and Hellman [DIF 1976] and independently by Merkle [MER 1979]. The first practical realization of a digital signature scheme appeared in the paper by Rivest, Shamir, and Adleman [RIV 1978].

Most introductory sources for digital signatures stress the need for digital signatures with message recovery coming from a public-key encryption system. Mitchell, Piper, and Wild [MIT 1992] give a good general treatment of the subject. Stinson [STI 1995] provides a similar elementary but general introduction. Many types of digital signatures with specific properties have been created, such as blind signatures, undeniable signatures, and fail stop signatures.

[TSE 2002] adopt the concept of self-certified public keys to propose a new signature scheme with message recovery. The proposed scheme has two properties that the signer's public key can simultaneously be authenticated in verifying the signature, and the receiver also obtains the message. As compared with the certificate-based signature scheme with message recovery, the public space and the communication cost are reduced.

[LEE 2002] proposed a generalized group-oriented threshold signature scheme and a generalized authenticated encryption scheme with shared verification.

The security of ordinary digital signature schemes relies on a computational assumption. Fail-stop signature schemes provide security for a sender against a forger with unlimited computational power by enabling the sender to provide a proof of forgery if it occurs. [SUS 2000] gives an efficient fail-stop signature scheme that uses two hard problems, discrete logarithm and factorization, as the basis of a receiver's security. The scheme has provable security against adaptively chosen message attack, and is the most efficient scheme with respect to the ratio of the message length to the signature length. The scheme provides an efficient solution to signing messages up to 1881 bits.

3.13 Authentication and Identification

Authentication is one of the most important of all information security objectives. Until the mid 1970s it was generally believed that secrecy and authentication were intrinsically connected. With the discovery of hash functions and digital signatures, it was realized that secrecy and authentication were truly separate and independent information security objectives. It may at first not seem important to separate the two but there are situations where it is not only useful but essential.

Much effort has been devoted to developing a theory of authentication. At the forefront of this is Simmons [SIM 1992], whose contributions

are nicely summarized by Massey [MAS 1992]. More concrete example of the necessity for authentication without secrecy, can be seen in the article by Simmons [SIM 1992 A].

An *identification* or *entity authentication* technique assures identity of the party involved, and confirms that he was active at the time the evidence was created or acquired.

3.14 Hash functions

One of the fundamental primitives in modern cryptography is the cryptographic hash function, often informally called a one-way hash function. A *hash function* is a computationally efficient function mapping binary strings of arbitrary length to binary strings of some fixed length, called *hash-values*. Much of the early work on cryptographic hash functions was done by Merkle [MER 1979]. The most comprehensive current treatment of the subject is by Preneel [PRE 1993].

3.15 Protocols and Mechanisms

A *cryptographic protocol (protocol)* is a distributed algorithm defined by a sequence of steps precisely specifying the actions required of two or more entities to achieve a specific security objective. As opposed to a protocol, a *mechanism* is a more general term encompassing protocols, algorithms (specifying the steps followed by a single entity), and non-cryptographic techniques (e.g., hardware protection and procedural controls) to achieve specific security objectives.

In order to ensure the security of the messages in computer networks, a new cryptographic protocol that uses electronic 'passports' has been developed [BET 1995]. The protocol which is called SELANE (secure local-area network) is compatible with almost every commercial and academic network.

A large number of successful cryptanalytic attacks on systems claiming security are due to protocol failure. An overview of this area is given by Moore [MOR 1992], including classifications of protocol failures and design principles.

Protocols play a major role in cryptography and are essential in meeting cryptographic goals. Encryption schemes, digital signatures, hash functions, and random number generation are among the primitives which may be utilized to build a protocol.

Protocols use basic functions to realize private communications on an unsecured channel. The basic primitives are the symmetric-key and the public-key encryption schemes. The protocols have shortcomings including the impersonation attack.

Often the role of public-key encryption in privacy communications is exactly the one suggested by this protocol. Public-key encryption is used as a means to exchange keys for subsequent use in symmetric-key encryption, motivated by performance differences between symmetric-key and public-key encryption.

A protocol failure or mechanism failure occurs when a mechanism fails to meet the goals for which it was intended, in a manner whereby an adversary gains advantage not by breaking an underlying primitive such as an encryption algorithm directly, but by manipulating the protocol or mechanism itself.

Protocols and mechanisms may fail for a number of reasons, including:

1. weaknesses in a particular cryptographic primitive which may be amplified by the protocol or mechanism
2. claimed or assumed security guarantees which are overstated or not clearly understood
3. the oversight of some principle applicable to a broad class of primitives such as encryption.

16 Key Establishment, Management and Certification

Key establishment is any process whereby a shared secret key becomes available to two or more parties, for subsequent cryptographic use ensuring the secured distribution of keys.

Key management is the set of processes and mechanisms which support key establishment and the maintenance of ongoing keying relationships between parties, including replacing older keys with new keys as necessary.

One approach to distributing public-keys is the so-called Merkle channel described in Simmons [SIM 1992]. Merkle proposed that public keys be distributed over so many independent public channels (newspaper, radio, television, etc.) that it would be improbable for an adversary to compromise all of them.

In 1978 Kohnfelder [KOH 1978] suggested the idea of using public-key certificates to facilitate the distribution of public keys over unsecured channels, such that their authenticity can be verified. The same idea proposed by Needham and Schroeder is explained in Wilkes [WIL 1975].

6.1 Trusted Third Parties (TTP) and Public Key Certificates

The trust placed on any entity varies with the way it is used, and hence motivates the following classification.

A TTP is said to be unconditionally trusted as if it is trusted on all matters. For example, it may have access to the secret and private keys of users. A TTP is said to be functionally trusted if the entity is assumed to be honest and fair but it does not have access to the secret or private keys of users.

A functionally trusted TTP could be used to register or certify users and contents of documents or as a judge. The distribution of public keys is generally easier than that of symmetric keys, since secrecy is not required. However, the integrity (authenticity) of public keys is

critical. A public key certificate consists of a *data part* and a *signature part*. The data part consists of the name of an entity, the public key corresponding to that entity, possibly additional relevant information (e.g., the entity's street or network address, a validity period for the public key, and various other attributes). The signature part consists of the signature of a TTP over the data part.

In order for an entity Y to verify the authenticity of the public key of an entity X , Y must have an authentic copy of the public key signature verification function of the TTP. For simplicity, assume that the authenticity of this verification function is provided to Y by non-cryptographic means. For example, if Y obtains it from the TTP in person, Y can then carry out the following steps.

1. Acquire the public-key certificate of X over some unsecured channel, either from a central database of certificates, or from X directly.
2. Use the TTP's verification function to verify the TTP's signature on X 's certificate.
3. If this signature verifies correctly, accept the public key in the certificate as X 's authentic public key, otherwise assume the public key is invalid.

Before creating a public-key certificate for X , the TTP must take appropriate measures to verify the identity of X and the fact that the public key to be certified actually belongs to X . One method is that X

appear before the TTP with a conventional passport as proof of identity, and obtain X 's public key from X in person along with evidence that X knows the corresponding private key. Once the TTP creates a certificate for a party, the trust that all other entities have in the authenticity of the TTP's public key can be used transitively to gain trust in the authenticity of that party's public key, through acquisition and verification of the certificate.

3.17 Pseudo Random Numbers and Sequences

Random number generation is an important primitive in many cryptographic mechanisms. For example, keys for encryption transformations need to be generated in a manner which is unpredictable to an adversary. Generating a random key involves the selection of random numbers or bit sequences. Random number generation presents challenging issues. Pseudorandom generators are fundamental to many theoretical and applied aspects of computing.

Knuth [KNU 1981] gives detailed treatment of many pseudorandom sequence generators. Knuth cites an example of a complex scheme to generate random numbers which on closer analysis is shown to produce numbers which are far from random.

Construction of a pseudorandom generator from any one-way function is described in [HAS 1999].

Certain applications in cryptography require the use of a truly Random Number Generator (RNG), a device which produces unpredictable and

unbiased digital signals derived from a fundamental noise mechanism. For IC-based cryptographic systems, an RNG must harness randomness from a low-power noise signal yet remain insensitive to deterministic influences such as crosstalk, power supply noise, and clock signal coupling through the substrate. In [PET 2000], design and fabrication techniques of an RNG IC utilizing established analog IC design techniques are explained.

A novel technique for random optical encoding is suggested in [ZAL 2000]. The proposed technique is based upon two binary masks: an encoding mask and a decoding mask. Each mask in itself is random, and contains no information that may be decoded. Only when the two masks are joined together, the decoded information is revealed. This way, the encoding of gray level as well as color information is possible. This approach is especially suitable for security applications.

Reflection or transmission of a quantum particle on a beam splitter is inherently random quantum process. [SOU 2001] presents an easy random number generator based on the division of weak light pulses on a beam splitter.

Since most of the random sequences come from physical means, they tend to be either costly or slow in their generation. To overcome these problems, methods have been devised to construct pseudorandom sequences in a deterministic manner from a shorter random sequence called a seed. The pseudorandom sequences appear to be generated by a truly random source to anyone not knowing the method of generation. Often the generation algorithm is known to all, but the seed

is unknown except to the entity generating the sequence. Many algorithms have been developed to generate pseudorandom bit sequences of various types. Many of these are completely unsuitable for cryptographic purposes and one must be cautious of claims by creators of such algorithms as to the random nature of the output.

[PAT 2002] shows how to create a provably secure block cipher, cryptographic pseudorandom generator, and pseudorandom function.

[NAO2002] presents an efficient construction of pseudo-random functions whose security is based on the intractability of factoring.

[WAN 2002] contrast the notions of complexity-theoretic pseudorandom strings (from algorithmic information theory) and pseudorandom strings (from cryptography).

CHAPTER IV

ME DEPENDANT MULTIPLE RANDOM CIPHER CODE

- 4.1 Introduction
- 4.2 Mandatory Requirement of Practical Encryption Systems
- 4.3 Structure of TDMRC Code
- 4.4 Algorithm of TDMRC Code
- 4.5 Key of TDMRC Code
- 4.6 Time Dependency
- 4.7 Polyalphabetic Nature
- 4.8 Pseudo Random Nature
- 4.9 Variable Block Length
- 4.10 Comparison with Other Conventional Schemes
- 4.11 Crypt Analysis of TDMRC Code
- 4.12 Test to check the vulnerability of TDMRC Code

Introduction

Before developing any new encryption method the deficiencies of the current systems are to be identified and the mandatory requirement of the new system is to be ascertained.

Fault Tolerant Hard Real Time Systems need data communication channels of high throughput. From the comparison of symmetric key systems and public key systems in Sec 3.8 it can be seen the Public key Encryption methods are inherently not so fast to meet this requirement. The key length is very big with the case of private key. Public key encryption techniques may be used to establish a key for a symmetric key system being used for further data transfer.

The keys in public key system can be used for long time without modification. We can take advantage of the long term nature of the public / private keys of the public key scheme and the performance efficiency of the symmetric key scheme. Since data encryption is frequently the most time consuming part of the encryption process, the public key scheme for key establishment is a small fraction of the total encryption process. The computational performance of public key encryption is inferior to that of symmetric key encryption.

Mandatory Requirement of Practical Encryption Systems

Mandatory requirements of any practical encryption systems are given by A. Kerchoffs in [KER 1883]. They are given below as Kerchoffs originally stated them.

1. the system should be, if not theoretically unbreakable, unbreakable in practice
2. compromise of the system details should not inconvenience the correspondents
3. the key should be rememberable without notes and can be easily changed
4. the cryptogram should be transmissible by telegraph
5. the encryption apparatus should be portable and operable by a single person
6. the system should be easy, requiring neither the knowledge of a long list of rules nor mental strain.

This list of requirements was articulated in 1883 and for the most part, remains useful today. Point 2 allows that the class of encryption transformations being used be publicly known and that the security of the system should reside only in the key chosen.

In 1972 the US Federal Department of Commerce took precautions to improve national security by calling for a cryptographic standard for storing, processing and distributing information, as a result of the increasing number of applications of computer systems. An appeal

was made for proposals for a cryptographic algorithm, the specifications of which were defined as follows:

- high level of security
- comprehensive and transparent specification
- security may not rely on the secrecy of the algorithm
- available and accessible to all users
- suitable for a variety of applications
- low cost implementation
- able to be exported
- accessible for validation.

3 Structure of Time Dependant Multiple Random Cipher Code

Time Dependant Multiple Random Cipher Code (TDMRC Code) proposed and developed in this work has three complexities. The details of complexities are given below.

1. TDMRC Code is time dependant. The codes used for any character differs depending upon time. Even for centi second difference, the codes will change.
2. It is poly alphabetic. The code used for the same character at different locations of the plain text are different. Poly Alphabetic Coefficient (PAC) decides the number of codes used corresponding to each plain text character.

3. It uses pseudo random number generation technique for code generation. Depending upon the random seed the codes will change.

TDMRC follows symmetric key method and uses less complex mathematical operations compared with any other schemes. It is a substitution coding system.

This method uses variable block length depending upon PAC where as the conventional methods are of fixed block length. And since many complexities are simultaneously incorporated TDMRC is a Product Code. In the case of TDMRC cryptanalysis is practically impossible. Though it is specifically designed for use in the communication channels of FTHRT system, it can be used for any other applications which requires data security.

4 Key of TDMRC Code

The Key of TDMRC Code consists of 3 elements.

1. Master Key derived from the Real Time Clock. It is an 8 digit number obtained by combining the values of hour, minute, second and centi second.
2. Poly Alphabetic Coefficient (PAC) which is actually a single digit number, P, indicating the number of codes simultaneously used for any character in an encrypting

session. This is to be decided at encryption stage. P can be any value, (need not be limited to single digit value) but a value of 3 will be sufficient to achieve computational security (explained in Section 4.11).

3. P number of 4 digit Sub Keys to be decided at the encryption stage.

Master Key can take 8640000 unique values in the range 00000000 to 99999999.

Similarly, Sub Keys can be any four numbers in the range 0000 to 9999

TDMRC Code is a symmetric key system. Hence the same key used at encryption stage itself is to be used for decryption also.

The Random Seeds for generation of codes for encryption and decryption can be generated by multiplying the Master Key with the Sub Keys and taking the 8 digits from extreme right of each product.

This will be more clear with the following example.

Assume Poly Alphabetic Coefficient (PAC) as 4, Real Time Clock Time as 11: 34 : 45.78 and the 4 Sub Keys as 2345, 4578, 1987, 1573

Now, Master Key is derived from RTC time simply by combining the two digit values of hour, minute, second and centi second. So Master Key in the current case is

11344578

First Random Seed is 8 digits from extreme right of the product
(11344578 X 2345 i.e. 03035410

Similarly,

Second Random Seed is 8 digits from extreme right of the product
(11344578 X 4578) i.e 35478084

Third Random Seed is 8 digits from extreme right of the product
(11344578 X 1987) i.e 41676486

Fourth Random seed is 8 digits from extreme right of the product
(11344578 X 1573) i.e 45021194

For a change of 1 centi second in the real time clock time, the corresponding changes in the random seed values will be of the order of thousands. This is clear from Table 4.1. In the table PAC is assumed as 4 and correspondingly 4 Sub Keys are also assumed. Starting from real time clock time of 11: 37: 45.78 onwards, for each incremental change of every 1 centi second, corresponding random seed values are calculated.

TABLE 4.1 - RANDOM SEEDS AT DIFFERENT TIMES

	MASTER KEY	SUB KEY 1	SUB KEY 2	SUB KEY 3	SUB KEY 4	RANDOM SEED 1	RANDOM SEED 2	RANDOM SEED 3	RANDOM SEED 4
78	11374578	2784	7823	4019	9107	66825152	83323694	14428982	88281846
79	11374579	2784	7823	4019	9107	66827936	83331517	14433001	88290953
80	11374580	2784	7823	4019	9107	66830720	83339340	14437020	88300060
81	11374581	2784	7823	4019	9107	66833504	83347163	14441039	88309167
82	11374582	2784	7823	4019	9107	66836288	83354986	14445058	88318274
83	11374583	2784	7823	4019	9107	66839072	83362809	14449077	88327381
84	11374584	2784	7823	4019	9107	66841856	83370632	14453096	88336488
85	11374585	2784	7823	4019	9107	66844640	83378455	14457115	88345595
86	11374586	2784	7823	4019	9107	66847424	83386278	14461134	88354702
87	11374587	2784	7823	4019	9107	66850208	83394101	14465153	88363809
88	11374588	2784	7823	4019	9107	66852992	83401924	14469172	88372916
89	11374589	2784	7823	4019	9107	66855776	83409747	14473191	88382023
90	11374590	2784	7823	4019	9107	66858560	83417570	14477210	88391130
91	11374591	2784	7823	4019	9107	66861344	83425393	14481229	88400237
92	11374592	2784	7823	4019	9107	66864128	83433216	14485248	88409344
93	11374593	2784	7823	4019	9107	66866912	83441039	14489267	88418451
94	11374594	2784	7823	4019	9107	66869696	83448862	14493286	88427558
95	11374595	2784	7823	4019	9107	66872480	83456685	14497305	88436665
96	11374596	2784	7823	4019	9107	66875264	83464508	14501324	88445772
97	11374597	2784	7823	4019	9107	66878048	83472331	14505343	88454879
98	11374598	2784	7823	4019	9107	66880832	83480154	14509362	88463986
99	11374599	2784	7823	4019	9107	66883616	83487977	14513381	88473093
100	11374600	2784	7823	4019	9107	66886400	83495800	14517400	88482200
01	11374601	2784	7823	4019	9107	66889184	83503623	14521419	88491307
02	11374602	2784	7823	4019	9107	66891968	83511446	14525438	88500414
03	11374603	2784	7823	4019	9107	66894752	83519269	14529457	88509521
04	11374604	2784	7823	4019	9107	66897536	83527092	14533476	88518628
05	11374605	2784	7823	4019	9107	66900320	83534915	14537495	88527735
06	11374606	2784	7823	4019	9107	66903104	83542738	14541514	88536842
07	11374607	2784	7823	4019	9107	66905888	83550561	14545533	88545949
08	11374608	2784	7823	4019	9107	66908672	83558384	14549552	88555056
09	11374609	2784	7823	4019	9107	66911456	83566207	14553571	88564163
10	11374610	2784	7823	4019	9107	66914240	83574030	14557590	88573270
11	11374611	2784	7823	4019	9107	66917024	83581853	14561609	88582377
12	11374612	2784	7823	4019	9107	66919808	83589676	14565628	88591484
13	11374613	2784	7823	4019	9107	66922592	83597499	14569647	88600591
14	11374614	2784	7823	4019	9107	66925376	83605322	14573666	88609698

It is evident that in TDMRC Coding system, only a centi second difference is enough to get an entirely different cipher text for any plain text.

4.5 Algorithm of TDMRC Code

The algorithm of TDMRC Code is given below for Encryption and Decryption separately.

Encryption Algorithm

- Step #1 Decide the number of codes that is to be used simultaneously ie. Poly alphabetic coefficient, P
- Step #2 Decide P number of sub keys, each key with 4 digits, $S_1S_1S_1S_1, S_2S_2S_2S_2, \dots, S_pS_pS_pS_p$
- Step #3 Read the Real Time Clock Time (System Time) with accuracy to centi second and form an 8 digit number TTTTTTTT. This will act as the Master Key.
- Step #4 Multiply the Master Key with the first Sub Key and take 8 digits of the product from extreme right to form the first Random Seed. Similarly, generate P number of Random Seeds.

Step #5 Generate P numbers of random series using the P numbers of random seeds generated in step #4, with 256 unique elements in each series. The elements should be of value 0 – 255 in decimal (00000000 – 11111111 in binary).

Step #6 Take data in blocks of P number of ASCII characters. Find the ASCII value of each character and substitute each character with element in the random series corresponding to this ASCII value. The first character in block of P characters is to be substituted with element from first random series, second character with element of second series and so on.

Decryption Algorithm

Step #1 Using the same keys used for encryption, regenerate P number of random seeds and P numbers of random series with 256 unique elements in each series. The elements should be of value 0-255 in decimal (00000000 – 11111111 in binary).

(The Pseudo Random Number Generation algorithm used should be same as the one used at encoding stage)

Step #2 Take cipher text in blocks of P number of ASCII characters. Find the ASCII value of each character and then substitute each character with the string character of

the serial number value of the element in the random series, the element which is same as the ASCII character in the block. The first character in block of P characters be substituted with the string character of the serial number value of the element from first random series, second character with the string character with the serial number value of the element from second random series and so on.

The algorithm of encryption and decryption using TDMRC Code is demonstrated in Plate #1 and Plate #2 respectively. Here PAC is taken as 3. Hence 3 codes are used simultaneously.

Time Dependency of TDMRC Code

Since random seed is derived from the Master Key which in turn depends on the real time clock of the computer, the series of random numbers generated will be different at each instant. In all operating systems, there will be facility to extract the real time clock timing. Usually it is available in milli second accuracy. Then, if the same passage itself is encoded many times using TDMRC Code, each time a new code and hence new cipher text is generated.

To study the time dependency nature of TDMRC Code, plain text, with sample data consisting of characters from alphabets, special character and numerals, is created. This plain text is encrypted at different times, keeping PAC and Sub Keys same.

PLAIN TEXT	THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS																										
PLAIN TEXT	T	H	E		Q	U	I	C	K		B	R	O	W	N		F	O	X		J						
Value	84	72	69	32	81	85	73	67	75	32	66	82	79	87	78	32	70	79	88	32	74						
Form Code No.	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3						
SP 32	51	110	58	51	110	58	51	110	58	51	110	58	51	110	58	51	110	58	51	110	58						
< 60	91	33	45	91	33	45	91	33	45	91	33	45	91	33	45	91	33	45	91	33	45						
= 61	57	124	121	57	124	121	57	124	121	57	124	121	57	124	121	57	124	121	57	124	121						
> 62	80	36	60	80	36	60	80	36	60	80	36	60	80	36	60	80	36	60	80	36	60						
? 63	46	79	108	46	79	108	46	79	108	46	79	108	46	79	108	46	79	108	46	79	108						
@ 64	68	106	32	68	106	32	68	106	32	68	106	32	68	106	32	68	106	32	68	106	32						
A 65	47	122	77	47	122	77	47	122	77	47	122	77	47	122	77	47	122	77	47	122	77						
B 66	62	107	125	62	107	125	62	107	125	62	107	125	62	107	125	62	107	125	62	107	125						
C 67	117	97	118	117	97	118	117	97	118	117	97	118	117	97	118	117	97	118	117	97	118						
D 68	64	44	93	64	44	93	64	44	93	64	44	93	64	44	93	64	44	93	64	44	93						
E 69	39	109	99	39	109	99	39	109	99	39	109	99	39	109	99	39	109	99	39	109	99						
F 70	61	72	34	61	72	34	61	72	34	61	72	34	61	72	34	61	72	34	61	72	34						
G 71	86	119	101	86	119	101	86	119	101	86	119	101	86	119	101	86	119	101	86	119	101						
H 72	75	92	71	75	92	71	75	92	71	75	92	71	75	92	71	75	92	71	75	92	71						
I 73	35	95	59	35	95	59	35	95	59	35	95	59	35	95	59	35	95	59	35	95	59						
J 74	37	105	70	37	105	70	37	105	70	37	105	70	37	105	70	37	105	70	37	105	70						
K 75	96	56	65	96	56	65	96	56	65	96	56	65	96	56	65	96	56	65	96	56	65						
L 76	41	42	103	41	42	103	41	42	103	41	42	103	41	42	103	41	42	103	41	42	103						
M 77	66	98	67	66	98	67	66	98	67	66	98	67	66	98	67	66	98	67	66	98	67						
N 78	50	78	111	50	78	111	50	78	111	50	78	111	50	78	111	50	78	111	50	78	111						
O 79	69	94	102	69	94	102	69	94	102	69	94	102	69	94	102	69	94	102	69	94	102						
P 80	55	43	90	55	43	90	55	43	90	55	43	90	55	43	90	55	43	90	55	43	90						
Q 81	123	85	81	123	85	81	123	85	81	123	85	81	123	85	81	123	85	81	123	85	81						
R 82	84	74	87	84	74	87	84	74	87	84	74	87	84	74	87	84	74	87	84	74	87						
S 83	89	112	82	89	112	82	89	112	82	89	112	82	89	112	82	89	112	82	89	112	82						
T 84	100	114	76	100	114	76	100	114	76	100	114	76	100	114	76	100	114	76	100	114	76						
U 85	115	116	48	115	116	48	115	116	48	115	116	48	115	116	48	115	116	48	115	116	48						
V 86	120	49	104	120	49	104	120	49	104	120	49	104	120	49	104	120	49	104	120	49	104						
W 87	38	52	40	38	52	40	38	52	40	38	52	40	38	52	40	38	52	40	38	52	40						
X 88	73	63	88	73	63	88	73	63	88	73	63	88	73	63	88	73	63	88	73	63	88						
Y 89	54	113	92	54	113	92	54	113	92	54	113	92	54	113	92	54	113	92	54	113	92						
Z 90	83	53	95	83	53	95	83	53	95	83	53	95	83	53	95	83	53	95	83	53	95						
Key of Cipher Code.	100	92	99	51	85	48	35	97	65	51	107	87	69	52	111	51	72	102	73	110	70						
OF ASCII	d	\	c	3	U	0	#	a	A	3	k	W	E	4	o	3	H	f	l	n	F						
RC TEXT	d\c3U0#aA3kWE4o3HfInFsbZ'~:ElcTnLKm:)z-6n]EwR																										

NOTE # 1 - EXAMPLE OF ENCRYPTION USING TDMRC CODE

RC TEXT → d\c3U0#aA3kWE4o3HfInFsbZ'`:ElcTnLkM:)z-6n]EwR

RC TEXT	d	l	c	3	U	0	#	a	A	3	k	W	E	4	o	3	H	f	i	n	F
Value	100	92	99	51	85	48	35	97	65	51	107	87	69	52	111	51	72	102	73	110	70
Form Code No.	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
SP	32	51	110	58	51	110	58	51	110	58	51	110	58	51	110	58	51	110	58	51	110
<	60	91	33	45	91	33	45	91	33	45	91	33	45	91	33	45	91	33	45	91	33
=	61	57	124	121	57	124	121	57	124	121	57	124	121	57	124	121	57	124	121	57	124
>	62	80	36	60	80	36	60	80	36	60	80	36	60	80	36	60	80	36	60	80	36
?	63	46	79	108	46	79	108	46	79	108	46	79	108	46	79	108	46	79	108	46	79
@	64	68	106	32	68	106	32	68	106	32	68	106	32	68	106	32	68	106	32	68	106
A	65	47	122	77	47	122	77	47	122	77	47	122	77	47	122	77	47	122	77	47	122
B	66	62	107	125	62	107	125	62	107	125	62	107	125	62	107	125	62	107	125	62	107
C	67	117	97	118	117	97	118	117	97	118	117	97	118	117	97	118	117	97	118	117	97
D	68	64	44	93	64	44	93	64	44	93	64	44	93	64	44	93	64	44	93	64	44
E	69	39	109	99	39	109	99	39	109	99	39	109	99	39	109	99	39	109	99	39	109
F	70	61	72	34	61	72	34	61	72	34	61	72	34	61	72	34	61	72	34	61	72
G	71	86	119	101	86	119	101	86	119	101	86	119	101	86	119	101	86	119	101	86	119
H	72	75	92	71	75	92	71	75	92	71	75	92	71	75	92	71	75	92	71	75	92
I	73	35	95	59	35	95	59	35	95	59	35	95	59	35	95	59	35	95	59	35	95
J	74	37	105	70	37	105	70	37	105	70	37	105	70	37	105	70	37	105	70	37	105
K	75	96	56	65	96	56	65	96	56	65	96	56	65	96	56	65	96	56	65	96	56
L	76	41	42	103	41	42	103	41	42	103	41	42	103	41	42	103	41	42	103	41	42
M	77	66	98	67	66	98	67	66	98	67	66	98	67	66	98	67	66	98	67	66	98
N	78	50	78	111	50	78	111	50	78	111	50	78	111	50	78	111	50	78	111	50	78
O	79	69	94	102	69	94	102	69	94	102	69	94	102	69	94	102	69	94	102	69	94
P	80	55	43	90	55	43	90	55	43	90	55	43	90	55	43	90	55	43	90	55	43
Q	81	123	85	81	123	85	81	123	85	81	123	85	81	123	85	81	123	85	81	123	85
R	82	84	74	87	84	74	87	84	74	87	84	74	87	84	74	87	84	74	87	84	74
S	83	89	112	82	89	112	82	89	112	82	89	112	82	89	112	82	89	112	82	89	112
T	84	100	114	76	100	114	76	100	114	76	100	114	76	100	114	76	100	114	76	100	114
U	85	115	116	48	115	116	48	115	116	48	115	116	48	115	116	48	115	116	48	115	116
V	86	120	49	104	120	49	104	120	49	104	120	49	104	120	49	104	120	49	104	120	49
W	87	38	52	40	38	52	40	38	52	40	38	52	40	38	52	40	38	52	40	38	52
X	88	73	63	88	73	63	88	73	63	88	73	63	88	73	63	88	73	63	88	73	63
Y	89	54	113	92	54	113	92	54	113	92	54	113	92	54	113	92	54	113	92	54	113
Z	90	83	53	95	83	53	95	83	53	95	83	53	95	83	53	95	83	53	95	83	53
of Plain Chr.	84	72	69	32	81	85	73	67	75	32	66	82	79	87	78	32	70	79	88	32	74
of ASCII	T	H	E		Q	U	I	C	K		B	R	O	W	N		F	O	X		J

PLAIN TEXT → THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS

NOTE #2 - EXAMPLE OF DECRYPTION USING TDMRC CODE

This can be seen from the following plates.

Plate #3 Plain Text Data consisting of characters from alphabets, special character and numerals

Plate #4 Encrypted Text at 11: 25 : 34.68 am
Master Key – 11253468
PAC – 5, Sub Keys – 1234, 2345, 3456, 4567 and 5678

Plate #5 Encrypted Text at 11: 45 : 42.39 am
Master Key – 11454239
PAC – 5, Sub Keys – 1234, 2345, 3456, 4567 and 5678

Plate #6 Encrypted Text at 11: 53 : 11.82 am
Master Key – 11531182
PAC – 5, Sub Keys – 1234, 2345, 3456, 4567 and 5678

Plate #7 Encrypted Text at 12 : 04 : 18.20 pm
Master Key – 12041820
PAC – 5, Sub Keys – 1234, 2345, 3456, 4567 and 5678

Plate #8 Encrypted Text at 1: 15 : 04.39 pm
Master Key – 13150439
PAC – 5, Sub Keys – 1234, 2345, 3456, 4567 and 5678

Plate #9 Encrypted Text at 1: 32 : 29.84pm
Master Key – 13322984

PAC – 5, Sub Keys – 1234, 2345, 3456, 4567 and 5678

Plate #10 Encrypted Text at 1: 53 : 14.42 pm

Master Key – 13531442

PAC – 5, Sub Keys – 1234, 2345, 3456, 4567 and 5678

Plate #11 Encrypted Text at 2 : 23 : 38.19 pm

Master Key – 14233819

PAC – 5, Sub Keys – 1234, 2345, 3456, 4567 and 5678

Plate #12 Encrypted Text at 2 : 53 : 27.79 pm

Master Key – 14532779

PAC – 5, Sub Keys – 1234, 2345, 3456, 4567 and 5678

In all the above cases PAC and the Sub Keys are kept unchanged but depending on the time at which encryption is done, Master Key varies. It can be seen that the cipher text in each instant is different confirming time dependency of TDMRC Code.

4.7 Poly Alphabetic Nature

The number of codes simultaneously used for encryption is decided by the poly alphabet coefficient, P. P can take any value. If it is larger the risk of crypt analysis will be low. Larger P will take more time for generation of codes at encryption stage and regeneration of code at decryption stage.

`Ocb`Ocb_`Ocb_`Ocb_`Ocb_`Ocb_`Ocb_`Ocb_`Ocb_`Ocb_`O
0@k`10@k`10@k`10@k`10@k`10@k`10@k`10@k`10@k`10@k`10@k`10@k
5|pke5|pke5|pke5|pke5|pke5|pke5|pke5|pke5|pke5|pke5|p
CcN*(CcN*(CcN*(CcN*(CcN*(CcN*(CcN*(CcN*(CcN*(CcN*(CcN
.mBtk.mBtk.mBtk.mBtk.mBtk.mBtk.mBtk.mBtk.mBtk.mBtk.mB
6oCO>6oCO>6oCO>6oCO>6oCO>6oCO>6oCO>6oCO>6oCO>6oC

ftv>Cftv>Cftv>Cftv>Cftv>Cftv>Cftv>Cftv>Cftv>Cftv>Cftv
E7qRVE7qRVE7qRVE7qRVE7qRVE7qRVE7qRVE7qRVE7qRVE7qRVE7q
|_Ydf|_Ydf|_Ydf|_Ydf|_Ydf|_Ydf|_Ydf|_Ydf|_Ydf|_Ydf|_Y
96Dxp96Dxp96Dxp96Dxp96Dxp96Dxp96Dxp96Dxp96Dxp96Dxp96D
XM/rHXM/rHXM/rHXM/rHXM/rHXM/rHXM/rHXM/rHXM/rHXM/rHXM/
t1TPnt1TPnt1TPnt1TPnt1TPnt1TPnt1TPnt1TPnt1TPnt1TPnt1T

'J:KE'J:KE'J:KE'J:KE'J:KE'J:KE'J:KE'J:KE'J:KE'J:KE'J:
*TjYy*TjYy*TjYy*TjYy*TjYy*TjYy*TjYy*TjYy*TjYy*TjYy*Tj
2eunA2eunA2eunA2eunA2eunA2eunA2eunA2eunA2eunA2eunA2eunA2e
s9W5Js9W5Js9W5Js9W5Js9W5Js9W5Js9W5Js9W5Js9W5Js9W5Js9W
<qS+'<qS+'<qS+'<qS+'<qS+'<qS+'<qS+'<qS+'<qS+'<qS+'<qS
Pu@w?Pu@w?Pu@w?Pu@w?Pu@w?Pu@w?Pu@w?Pu@w?Pu@w?Pu@w?Pu@

V}a/ZV}a/ZV}a/ZV}a/ZV}a/ZV}a/ZV}a/ZV}a/ZV}a/ZV}a/ZV}a/ZV}a
\$p*Cg\$p*Cg\$p*Cg\$p*Cg\$p*Cg\$p*Cg\$p*Cg\$p*Cg\$p*Cg\$p*Cg\$p*Cg\$p*
v.'!8v.'!8v.'!8v.'!8v.'!8v.'!8v.'!8v.'!8v.'!8v.'!8v.'
FXmewFXmewFXmewFXmewFXmewFXmewFXmewFXmewFXmewFXmewFXm
?F1-..?F1-..?F1-..?F1-..?F1-..?F1-..?F1-..?F1-..?F1-..?F1
NQU2GNQU2GNQU2GNQU2GNQU2GNQU2GNQU2GNQU2GNQU2GNQU2GNQU

@]sS4>hYUQefn|I8snxQ[v\$hRrNn{RYNR}R8xvPH8Kn(=

RTC TIME - 11:25:34.68 am
MAIN KEY - 11253468
P A C - 5
SUB KEYS - 1234, 2345, 3456, 4567 and 5678

PLATE #4 - CIPHER TEXT AT 11:25:34.68 am

It can be seen that the time required for crypt analysis is geometrically proportional to the Poly Alphabetic Coefficient, P.

To study the poly alphabetic nature, sample data is encrypted many times with different P, keeping Master Key and Sub Keys same. From the attached Plates #11 to #20 it can be seen that for the same plain character we get different cipher characters depending on the poly alphabetic coefficient.

Plate #13	Sample Data file is created taking characters from alphabets, special characters and numerals.
Plate #14	The sample data file is encrypted using TDMRC Code. Master Key 11234567, PAC is 1, and Sub Key 2345.
Plate #15	PAC is taken as 2 and the same sample data is encoded using two codes.
Plate #16	Cipher Text with PAC 3
Plate #17	Cipher Text with PAC 4
Plate #18	Cipher Text with PAC 5
Plate #19	Cipher Text with PAC 10
Plate #20	Cipher Text with PAC 20
Plate #21	Cipher Text with PAC 30
Plate #22	Cipher Text with PAC 40
Plate #23	Cipher Text with PAC 50
Plate #24	Cipher Text with PAC 56

In all the above cases, the Master Key and Sub Keys are retained same; only the PAC has been changed. In each case, different cipher

X&fDX&fDX&fDX&fDX&fDX&fDX&fDX&fDX&fDX&fDX&fDX&fDX&fDX&fDX
tIsEtIsEtIsEtIsEtIsEtIsEtIsEtIsEtIsEtIsEtIsEtIsEtIsEt
HIvyHIvyHIvyHIvyHIvyHIvyHIvyHIvyHIvyHIvyHIvyHIvyHIvyHIvyH
s[6bs[6bs[6bs[6bs[6bs[6bs[6bs[6bs[6bs[6bs[6bs[6bs[6bs[6bs[6bs[6bs
<%w)<%w)<%w)<%w)<%w)<%w)<%w)<%w)<%w)<%w)<%w)<%w)<%w)<
q-e&q-e&q-e&q-e&q-e&q-e&q-e&q-e&q-e&q-e&q-e&q-e&q-e&q-e&q-e&q-e&q

nm4<nm4<nm4<nm4<nm4<nm4<nm4<nm4<nm4<nm4<nm4<nm4<nm4<nm4<n
mAG]mAG]mAG]mAG]mAG]mAG]mAG]mAG]mAG]mAG]mAG]mAG]mAG]mAG]mAG]m
kP2JkP2JkP2JkP2JkP2JkP2JkP2JkP2JkP2JkP2JkP2JkP2JkP2JkP2JkP2Jk
1K}a1K}a1K}a1K}a1K}a1K}a1K}a1K}a1K}a1K}a1K}a1K}a1K}a1K}a1
5RaY5RaY5RaY5RaY5RaY5RaY5RaY5RaY5RaY5RaY5RaY5RaY5RaY5RaY5RaY5
'nCA`nCA`nCA`nCA`nCA`nCA`nCA`nCA`nCA`nCA`nCA`nCA`nCA`nCA`nCA`nCA`

0V^_0V^_0V^_0V^_0V^_0V^_0V^_0V^_0V^_0V^_0V^_0V^_0V^_0V^_0V^_0V^_0
YaRtYaRtYaRtYaRtYaRtYaRtYaRtYaRtYaRtYaRtYaRtYaRtYaRtYaRtYaRtYaRtY
P||VP||VP||VP||VP||VP||VP||VP||VP||VP||VP||VP||VP||VP||VP||VP
Eu8\$Eu8\$Eu8\$Eu8\$Eu8\$Eu8\$Eu8\$Eu8\$Eu8\$Eu8\$Eu8\$Eu8\$Eu8\$Eu8\$Eu8\$E
S0%'S0%'S0%'S0%'S0%'S0%'S0%'S0%'S0%'S0%'S0%'S0%'S0%'S0%'S
N\$/BN\$/BN\$/BN\$/BN\$/BN\$/BN\$/BN\$/BN\$/BN\$/BN\$/BN\$/BN\$/BN\$/BN\$/BN

ftK>ftK>ftK>ftK>ftK>ftK>ftK>ftK>ftK>ftK>ftK>ftK>ftK>ftK>f
Ax! Ax! Ax! Ax! Ax! Ax! Ax! Ax! Ax! Ax! Ax! Ax! Ax! Ax! Ax! A
Rh*+Rh*+Rh*+Rh*+Rh*+Rh*+Rh*+Rh*+Rh*+Rh*+Rh*+Rh*+Rh*+Rh*+Rh*+R
v+F@v+F@v+F@v+F@v+F@v+F@v+F@v+F@v+F@v+F@v+F@v+F@v+F@v+F@v+F@v
M6PiM6PiM6PiM6PiM6PiM6PiM6PiM6PiM6PiM6PiM6PiM6PiM6PiM6PiM6PiM
02cf02cf02cf02cf02cf02cf02cf02cf02cf02cf02cf02cf02cf02cf02cf0
TOVmTOVmTOVmTOVmTOVmTOVmTOVmTOVmTOVmTOVmTOVmTOVmTOVmTOVmTOVmT

RTC TIME - 11:23:34.67 am
MASTER KEY - 11234567
P A C - 4
SUB KEYS - 2345, 3456, 4567, 5678

PLATE #17 - CIPHER TEXT with PAC = 4

X&fD\^B6H3X&fD\^B6H3X&fD\^B6H3X&fD\^B6H3X&fD\^B6H3X&f
t1sEkh [>3Dt1sEkh [>3Dt1sEkh [>3Dt1sEkh [>3Dt1sEkh [>3Dt1s
HIvyyixIN?HIvyyixIN?HIvyyixIN?HIvyyixIN?HIvyyixIN?HIv
s [6b) WU] >Ys [6b) WU] >Ys [6b) WU] >Ys [6b) WU] >Ys [6
<%w) 3v^&zy<%w) 3v^&zy<%w) 3v^&zy<%w) 3v^&zy<%w) 3v^&zy<%w
q-e&}dXNZGq-e&}dXNZGq-e&}dXNZGq-e&}dXNZGq-e&}dXNZGq-e

nm4<bKGg=Vnm4<bKGg=Vnm4<bKGg=Vnm4<bKGg=Vnm4<bKGg=Vnm4
mAG] Kz@`9zmAG] Kz@`9zmAG] Kz@`9zmAG] Kz@`9zmAG] Kz@`9zmAG
kP2JznEfFrkP2JznEfFrkP2JznEfFrkP2JznEfFrkP2JznEfFrkP2
1K}a<a8u) T1K}a<a8u) T1K}a<a8u) T1K}a<a8u) T1K}a<a8u) T1K}
5RaYvSw%XR5RaYvSw%XR5RaYvSw%XR5RaYvSw%XR5RaYvSw%XR5Ra
`nCAC+y5-B`nCAC+y5-B`nCAC+y5-B`nCAC+y5-B`nCAC+y5-B`nC

0V^_||Iz?h0V^_||Iz?h0V^_||Iz?h0V^_||Iz?h0V^_||Iz?h0V^
YaRtN>u}I\YaRtN>u}I\YaRtN>u}I\YaRtN>u}I\YaRtN>u}I\YaR
P||V8mPJr/P||V8mPJr/P||V8mPJr/P||V8mPJr/P||V8mPJr/P||
Eu8\$`t q Eu8\$`t q Eu8\$`t q Eu8\$`t q Eu8\$`t q Eu8
S0%'e_T2K_S0%'e_T2K_S0%'e_T2K_S0%'e_T2K_S0%'e_T2K_S0%
N\$/BtD\mp#N\$/BtD\mp#N\$/BtD\mp#N\$/BtD\mp#N\$/BtD\mp#N\$/

ftK>%'*E!.ftK>%'*E!.ftK>%'*E!.ftK>%'*E!.ftK>%'*E!.ftK
Ax! :qzLLpAx! :qzLLpAx! :qzLLpAx! :qzLLpAx! :qzLLpAx!
Rh*+n7) v\$=Rh*+n7) v\$=Rh*+n7) v\$=Rh*+n7) v\$=Rh*+n7) v\$=Rh*
v+F@`p]^ {wv+F@`p]^ {wv+F@`p]^ {wv+F@`p]^ {wv+F@`p]^ {wv+F
M6Pi_G {BQ [M6Pi_G {BQ [M6Pi_G {BQ [M6Pi_G {BQ [M6Pi_G {BQ [M6P
02cf] ovWwg02cf] ovWwg02cf] ovWwg02cf] ovWwg02cf] ovWwg02c
TOVm (fg: jcTOVm (fg: jcTOVm (fg: jcTOVm (fg: jcTOVm (fg: jcTOV

RTC TIME - 11:23:34.67 am
MASTER KEY - 11234567
P A C - 10
SUB KEYS - 2345, 3456, 4567, 5678, 6789,
7890, 8901, 9012, 0123, 1234

PLATE #19 - CIPHER TEXT with PAC = 10

X&fD\^B6H301u2RkkbsnX&fD\^B6H301u2RkkbsnX&fD\^B6H301u
tlseKh [>3DC.TPLGNOL\^tlseKh [>3DC.TPLGNOL\^tlseKh [>3DC.T
HIvyyixIN?#^91T/vB)YHIvyyixIN?#^91T/vB)YHIvyyixIN?#^9
s[6b)WU]>Y@GX*9? &B7s[6b)WU]>Y@GX*9? &B7s[6b)WU]>Y@GX
<%w)3v^&zyLMEbK-x)10<%w)3v^&zyLMEbK-x)10<%w)3v^&zyLME
q-e&}dXNZGrVo5MwMYTJq-e&}dXNZGrVo5MwMYTJq-e&}dXNZGrVo

nm4<bKGg=VA _H=[H*PNnm4<bKGg=VA _H=[H*PNnm4<bKGg=VA _
mAG]Kz@`9z|Ae`{'Bx%VmAG]Kz@`9z|Ae`{'Bx%VmAG]Kz@`9z|Ae
kP2JznEfFr6= VqCX+`wkP2JznEfFr6= VqCX+`wkP2JznEfFr6=
1K}a<a8u)T5/N{6{*.(p1K}a<a8u)T5/N{6{*.(p1K}a<a8u)T5/N
5RaYvSw%Xr1>=N`!@'gh5RaYvSw%Xr1>=N`!@'gh5RaYvSw%Xr1>=
`nCAC+y5-BNOfd'r2#|j`nCAC+y5-BNOfd'r2#|j`nCAC+y5-BNOf

0V^_|}Iz?h{21_.D\X\#0V^_|}Iz?h{21_.D\X\#0V^_|}Iz?h{21
YaRtN>u}I\KH|9BNW`9-YaRtN>u}I\KH|9BNW`9-YaRtN>u}I\KH|
P||V8mPJr/hipSD%cA6QP||V8mPJr/hipSD%cA6QP||V8mPJr/hip
Eu8\$jt q &rc6/1+HU+Eu8\$jt q &rc6/1+HU+Eu8\$jt q &rc
S0%'e_T2K_!Z!mJuGC=SS0%'e_T2K_!Z!mJuGC=SS0%'e_T2K_!Z!
N\$/BtD\mp#nsMe_TIoC!N\$/BtD\mp#nsMe_TIoC!N\$/BtD\mp#nsM

ftK>%*E!.=q>g3&bMD ftK>%*E!.=q>g3&bMD ftK>%*E!.=q>
Ax! :qzLLpjF?wEd}fh:Ax! :qzLLpjF?wEd}fh:Ax! :qzLLpjF?
Rh*+n7)v\$=/T(E1Un){FRh*+n7)v\$=/T(E1Un){FRh*+n7)v\$=/T(
v+F@`p)^{wpaFXjmP/M}v+F@`p)^{wpaFXjmP/M}v+F@`p)^{wpaF
M6Pi_G{BQ[ao-37W/QFqM6Pi_G{BQ[ao-37W/QFqM6Pi_G{BQ[ao-
02cf]ovWwg45HZ[7>=x<02cf]ovWwg45HZ[7>=x<02cf]ovWwg45H
TOVm(fg:jcg(I<g6r1cmTOVm(fg:jcg(I<g6r1cmTOVm(fg:jcg(I

RTC TIME - 11:23:34.67 am
MASTER KEY - 11234567
P A C - 20
SUB KEYS - 2345, 3456, 4567, 5678, 6789,
7890, 8901, 9012, 0123, 1234,
2456, 3567, 4678, 5789, 6890,
7901, 8012, 9123, 0234, 1345

PLATE #20 - CIPHER TEXT with PAC = 20

X&fD\^B6H301u2RkkbsnxQ!7\Drh#jX&fD\^B6H301u2RkkbsnxQ!
t1sEkh[>3DC.TPLGNOL\9jJt]o!cnFt1sEkh[>3DC.TPLGNOL\9jJ
HIvyyixIN?#^91T/vB)Yc#++#+?)'N2HIvyyixIN?#^91T/vB)Yc#+
s[6b)WU]>Y@GX*9? &B7\$OZZhZZ=-vs[6b)WU]>Y@GX*9? &B7\$OZ
<%w)3v^&zyLMEbK-x)10Ln9'(X>}U'<%w)3v^&zyLMEbK-x)10Ln9
q-e&}dXNZGrVo5MwMYTJNT/mB#cYoMq-e&}dXNZGrVo5MwMYTJNT/

nm4<bKGg=VA _H=[H*PNpa_wHH[(5Qnm4<bKGg=VA _H=[H*PNpa_
mAG]Kz@`9z|Ae`{'Bx%Vu.S!dYP`80mAG]Kz@`9z|Ae`{'Bx%Vu.S
kP2JznEfFr6= VqCX+`wJSn5acq\$` kP2JznEfFr6= VqCX+`wJSn
1K}a<a8u)T5/N{6{*.(pry7Ii>%NS11K}a<a8u)T5/N{6{*.(pry7
5RaYvSw%XR1>=N`!@'gh?BCMxq+{qH5RaYvSw%XR1>=N`!@'gh?BC
`nCAC+y5-BNOfd'r2#|jW?[UqQ^5>u`nCAC+y5-BNOfd'r2#|jW?[

0V^_||}Iz?h{21_.D\X\#=2L4*K:r1T0V^_||}Iz?h{21_.D\X\#=2L
YaRtN>u}I\KH|9BNW`9-MgpY.8J\ [DYaRtN>u}I\KH|9BNW`9-Mgp
P||V8mPJr/hipSD%cA6Qzr*dVf :07P||V8mPJr/hipSD%cA6Qzr*
Eu8\$j`t q &rc6/1+HU+&h|u%%YxC*Eu8\$j`t q &rc6/1+HU+&h|
S0%'e_T2K_!Z!mJuGC=S4\$(Vz\`T:/S0%'e_T2K_!Z!mJuGC=S4\$(
N\$/BtD\mp#nsMe_TIoC!#!m)yjXnEgN\$/BtD\mp#nsMe_TIoC!#!m

ftK>%'*E!.=q>g3&bMD {-EA)hkDX|ftK>%'*E!.=q>g3&bMD {-E
Ax! :qzLLpjF?wEd}fh: AwgM@9&K9Ax! :qzLLpjF?wEd}fh: Aw
Rh*+n7)v\$=/T(E1Un){FwGQBxOAEGBRh*+n7)v\$=/T(E1Un){FwGQ
v+F@`p]^ {wpaFXjMP/M}n&^ {^B@Iz\$v+F@`p]^ {wpaFXjMP/M}n&^
M6Pi_G {BQ[ao-37W/QFq6WAOb144kVM6Pi_G {BQ[ao-37W/QFq6WA
O2cf}ovWwg45HZ[7>=x<kf\$>?&UKZnO2cf}ovWwg45HZ[7>=x<kf\$
TOVm(fg:jcg(I<g6r1cm:{FFtx/l|WTOVm(fg:jcg(I<g6r1cm:{F

RTC TIME - 11:23:34.67 am
MASTER KEY - 11234567
P A C - 30
SUB KEYS - 2345, 3456, 4567, 5678, 6789,
7890, 8901, 9012, 0123, 1234,
2456, 3567, 4678, 5789, 6890,
7901, 8012, 9123, 0234, 1345,
2567, 3678, 4789, 5890, 6901,
7012, 8123, 9234, 0345, 1456

PLATE #21 - CIPHER TEXT with PAC = 30

X&fD\^B6H301u2RkkbsnxQ!7\Drh#jnyep<blTy/X&fD\^B6H301u
t1sEkh[>3DC.TPLGNOL\9jJt]o!cnF@-\Oi1*UXDt1sEkh[>3DC.T
HIvyyixIN?#^91T/vB)Yc#++#?)'N2:PBCYVrpTJHIvyyixIN?#^9
s[6b)WU]>Y@GX*9? &B7\$OZZhZZ=-v4mf|\`Ws_27s[6b)WU]>Y@GX
<%w)3v^&zyLMEbK-x)10Ln9'(X>}U'9Fd^t!AX/T<%w)3v^&zyLME
q-e&}dXNZGrVo5MwMYTJNT/mB#cYoM8J.2N6{\$^ q-e&}dXNZGrVo

nm4<bKGg=VA _H=[H*PNpa_wHH[(5QI1DtsJ/SN{nm4<bKGg=VA _
mAG]Kz@`9z|Ae`{'Bx%Vu.S!dYP`80W0Q1I?j?\${mAG]Kz@`9z|Ae
kP2JznEfr6=VqCX+`wJSn5acq\$`zxEs.=}H#ikP2JznEfr6=
1K}a<a8u)T5/N{6{*.(pry7Ii>%NSlxMV`e'\$Mp&1K}a<a8u)T5/N
5RaYvSw%Xr1>=N`!@'gh?BCMxq+{qHU\=c|)Jw=K5RaYvSw%Xr1>=
`nCAC+y5-BNOfd'r2#|jW?[UqQ^5>u?{i?rzc:Y`nCAC+y5-BNOfd

0V^_|}Iz?h{21_.D\X\#=2L4*K:r1T`<M00t2]!M0V^_|}Iz?h{21
YartN>u}I\KH|9BNW`9-MgpY.8J\[DKk)\OSbvneYartN>u}I\KH|
P|V8mPJr/hipSD%ca6Qzr*dvf :07s`}kA*dZ9:P|V8mPJr/hip
Eu8\$j`t q &rc6/1+HU+&h|u%YxC*ZU7evPB[80Eu8\$j`t q &rc
S0%'e_T2K_!Z!mJuGC=S4\$(Vz\`T:/3LgE\+%h6VS0%'e_T2K_!Z!
N\$/BtD\mp#nsMe_TIoC!#!m)yjXnEgb.^8:F(^B-N\$/BtD\mp#nsM

ftK>%'*E!.=q>g3&bMD {-EA)hkDX|}b/S'9G2@oftK>%'*E!.=q>
Ax! :qzLLpjF?wEd}fh: AwgM@9&K9] !qCT_d(8Ax! :qzLLpjF?
Rh*+n7)v\$=/T(E1Un){FwGQBxOAEGBs&NP}_w4G!Rh*+n7)v\$=/T(
v+F@`p]^ {wpaFXjMP/M}n&^ {^B@Iz\$H3099c7DEtv+F@`p]^ {wpaF
M6Pi_G {BQ[ao-37W/QFq6WAOb144kV>Q<d&|e|sLM6Pi_G {BQ[ao-
02cf]ovWwg45HZ[7>=x<kf\$>?&UKZn[%F:na@f))O2cf]ovWwg45H
TOVm(fg:jcg(I<g6r1cm:{FFtx/l|W.GP\$05IPm5TOVm(fg:jcg(I

- RTC TIME - 11:23:34.67 am
- MASTER KEY - 11234567
- P A C - 40
- SUB KEYS - 2345, 3456, 4567, 5678, 6789,
7890, 8901, 9012, 0123, 1234,
2456, 3567, 4678, 5789, 6890,
7901, 8012, 9123, 0234, 1345,
2567, 3678, 4789, 5890, 6901,
7012, 8123, 9234, 0345, 1456,
2678, 3789, 4890, 5901, 6012,
7123, 8234, 9345, 0456, 1567

PLATE #22 - CIPHER TEXT with PAC = 40

X&fD\^B6H301u2RkkbsnxQ!7\Drh#jnyep<blTy/V'Bp=*Ms5hX&f
tlseKh[>3DC.TPLGNOL\9jJt]o!cnF@-\Oi1*UXDv-W+oT`J`Ptls
HIvyyixIN?#^91T/vB)Yc#++#?)'N2:PBCYVrpTJ:yrWq9A{=>Hiv
s[6b)WU]>Y@GX*9? &B7\$OZZhZZ=-v4mf|\`Ws_27.s@@t&I`>zs[6
<%w)3v^&zyLMEbK-x)10Ln9'(X>}U'9Fd^t!AX/TH I'H'g/B/<%w
q-e&}dXNZGrVo5MwMYTJNT/mB#cYoM8J.2N6{\$^ >3ePFvlu-wq-e

nm4<bKGg=VA _H=[H*PNpa_wHH[(5QI1DtsJ/SN{2?T%!t2R'Wnm4
mAG]Kz@`9z|Ae`{'Bx%Vu.S!dYP`80W0Q1I?j?\${McC8}6}NaHmAG
kP2JznEfr6= VqCX+`wJSn5acq\$` zxEs.=}H#i#vS</|Jf<skP2
1K}a<a8u)T5/N{6{*.(pry7Ii>%NSlxMV`e'\$Mp&!6K4Nd{=bK1K}
5RaYvSw%Xr1>=N`!@'gh?BCMxq+{qHU\=c|)Jw=Kiiqxc#\$D@A5Ra
`nCAc+y5-BNOfd'r2#|jW?{UqQ^5>u?{i?rzc:YBaf*RgD|Le`nC

0V^_||Iz?h{21_.D\X\#=2L4*K:r1T`<M00t2]!M)]3Jv09L4]0V^
YaRtN>u}I\KH|9BNW`9-MgpY.8J\[DKk)\OSbvneqZ'v^3VaIuYaR
P||V8mPJr/hipSD%ca6Qzr*dvf :07s`}kA*dZ9:X/gj9Bemf:P||
Eu8\$jt q &rc6/1+HU+&h|u%%YxC*ZU7evPB[8OS(&aCzTTt%Eu8
S0%'e_T2K_!Z!mJuGC=S4\$(Vz\`T:/3LgE\+%h6V'V|HuA:#i!S0%
N\$/BtD\mp#nsMe_TIoC!#!m)yjXnEgb.^8:F(^B-?E%r)>%?/&N\$/

ftK>%'*E!.=q>g3&bMD {-EA)hkDX|}b/S'9G2@o1m[\psE4vDftK
Ax! :qzLLpjF?wEd}fh: AwgM@9&K9] !qCT_d(8<qwSjh)p{?Ax!
Rh*+n7)v\$=/T(E1Un){FwGQBxOAEGBs&NP}_w4G!0#p2r1*%j_Rh*
v+F@`p}^ {wpaFXjMP/M}n&^ {^B@Iz\$H3O99c7DEt5e-`QxX7pqv+F
M6Pi_g{BQ[ao-37W/QFq6WAOb144kV>Q<d&|e|sLhSZ1Z^hek+M6P
02cf]ovWwg45HZ[7>=x<kf\$>?&UKZn[%F:na@f))&2H3[-b8_rO2c
TOVm(fg:jcg(I<g6r1cm:{FFtx/l|W.GP\$05IPm5RghcV.^c[(TOV

RTC TIME - 11:23:34.67 am
MASTER KEY - 11234567
P A C - 50
SUB KEYS - 2345, 3456, 4567, 5678, 6789, 7890, 8901,
9012, 0123, 1234, 2456, 3567, 4678, 5789,
6890, 7901, 8012, 9123, 0234, 1345, 2567,
3678, 4789, 5890, 6901, 7012, 8123, 9234,
0345, 1456, 2678, 3789, 4890, 5901, 6012,
7123, 8234, 9345, 0456, 1567, 2789, 3890,
4901, 5012, 6123, 7234, 8345, 9456, 0567,
1678

PLATE #23 - CIPHER TEXT with PAC = 50

X&fD\^B6H301u2RkkbsnxQ!7\Drh#jnyep<blTy/V'Bp=*Ms5hu) P
tIsEkh[>3DC.TPLGNOL\9jJt]o!cnF@-\Oi1*UXDv-W+oT`J`P[hu
HIvyyixIN?#^91T/vB)Yc#++#?)'N2:PBCYVrpTJ:yrWq9A{=>]D{
s[6b)WU]>Y@GX*9? &B7\$OZZhZZ-=v4mf|`Ws_27.s@@t&I`>z V2
<%w)3v^&zyLMEbK-x)10Ln9'(X>}U'9Fd^t!AX/TH I]H'g/B//x'
q-e&}dXNZGrVo5MwMYTJNT/mB#cYoM8J.2N6{\$^ >3ePFvlu-wSe9

nm4<bKGG=VA _H=[H*PNpa_wHH[(5QI1DtsJ/SN{2?T%!t2R'WP&j
mAG]Kz@`9z|Ae`{'Bx%Vu.S!dYP`80W0Q1I?j?\${McC8}6}NaH.[f
kP2JznEfr6=VqCX+`wJSn5acq\$`zxEs.=}H#i#vS</|Jf<ssO(
1K}a<a8u)T5/N{6{*.(pry7Ii>%NSlxMV`e'\$Mp&!6K4Nd{=bKL\$4
5RaYvSw%Xr1>=N`!@'gh?BCMxq+{qHU\=c|)Jw=Kiiqxc#\$D@A\MA
`nCAc+y5-BNOfd'r2#|jW?[UqQ^5>u?{i?rzc:YBaf*RgD|Le`Cp

0V^_}|Iz?h{21_.D\X\#=2L4*K:r1T`<M00t2]!M)]3Jv09L4]Y%q
YaRtN>u}I\KH|9BNW`9-MgpY.8J\{DKk)\OSbvneqZ'v^3VaIu|Q}
P||V8mPJr/hipSD%cA6Qzr*dvf :07s`}kA*dZ9:X/gj9Bemf:%I?
Eu8\$J`t q &rc6/1+HU+&h|u%YxC*ZU7evPB[8OS(&aCzTTt%iE3
S0%'e_T2K_!Z!mJuGC=S4\$(Vz\`T:/3LgE\+%h6V'V|HuA:#i!(d!
N\$/BtD\mp#nsMe_TIoC!#!m)yjXnEgb.^8:F(^B-?E%r]>%?/&=G.

ftK>%'*E!.=q>g3&bMD {-EA)hkDX|}b/S'9G2@o1m[\pse4vDXkW
Ax! :qzLLpjF?wEd}fh: AwgM@9&K9] !qCT_d(8<qwSjh)p{?>nK
Rh*+n7)v\$=/T(E1Un){FwGQBxOAEGBs&NP}_w4G!0#p2r1*%j_5-\$
v+F@`p}^ {wpaFXjmP/M}n&^ {^B@Iz\$H3099c7DEt5e-`QxX7pqFXZ
M6Pi_G {BQ[ao-37W/QFq6WAOb144kV>Q<d&|e|sLhSZ1Z^hek+c9N
02cf]ovWwg45HZ[7>=x<kf\$>?&UKZn[%F:na@f))&2H3[-b8_rI:+
TOVm(fg:jcg(I<g6r1cm:{FFtx/l|W.GP\$05IPm5RghcV.^c[(zF*

RTC TIME - 11:23:34.67 am
MASTER KEY - 11234567
P A C - 56
SUB KEYS - 2345, 3456, 4567, 5678, 6789, 7890, 8901,
9012, 0123, 1234, 2456, 3567, 4678, 5789,
6890, 7901, 8012, 9123, 0234, 1345, 2567,
3678, 4789, 5890, 6901, 7012, 8123, 9234,
0345, 1456, 2678, 3789, 4890, 5901, 6012,
7123, 8234, 9345, 0456, 1567, 2789, 3890,
4901, 5012, 6123, 7234, 8345, 9456, 0567,
1678, 2890, 3901, 4012, 5123, 6234, 7345.

PLATE #24 - CIPHER TEXT with PAC = 56

text has been generated and it confirms that TDMRC Coding system is poly alphabetic in nature.

4.8 Pseudo Random Nature

In TDMRC Code any Random Number Generator (RNG), which ensures unique random series corresponding to any random seed, can be used. The codes generated at any instant depend upon the random series which in turn depend on the random seed. The random seed also depends upon centi second of real time. So a true random nature of code is assured.

In the experimental setup, random series generated based on different random seeds are given in the following Plates. It was so programmed that numbers in the range 0 to 255 are the elements of the series and any element will not repeat.

Plates # 25 Random seed is 00000000. Series starts with the number 174 and ends with 82

Plates # 26 Random seed is 00000001. Series starts with the number 91 and ends with 122.

Plates # 27 Random seed is 11111111. Series starts with the number 0 and ends with 92.

Plates # 28 Random seed is 11111112. Series starts with the number 144 and ends with 73.

- Plates #29 Random seed is 33333331. Series starts with the number 242 and ends with 216.
- Plates # 30 Random seed is 33333332. Series starts with the number 137 and ends with 106.
- Plates # 31 Random seed is 33333333. Series starts with the number 32 and ends with 182.
- Plates #32 Random seed is 33333334. Series starts with the number 202 and ends with 149.
- Plates # 33 Random seed is 99999998. Series starts with the number 198 and ends with 109.
- Plates # 34 Random seed is 99999999. Series starts with the number 6 and ends with 61.

In all the above cases the random series is different. It can be seen that a difference of 1 in the random seed value is enough to get an entirely different random series. It confirms that TDMRC Code generated at any instant is pseudo random in nature.

4.9 Variable Block Length

In this method of encryption P number of ASCII characters (8 bit units) are considered at a time; P represent the Poly Alphabetic Coefficient. The value of P can be decided / modified at the beginning

174	211	182	68	104	72	173	154	241	50	230		
196	233	44	124	177	83	151	125	37	206	226		
172	70	150	180	255	57	110	148	134	114	153		
205	231	138	143	32	129	79	127	214	17	207		
183	71	176	93	14	128	136	140	252	116	21	187	
249	204	220	198	170	16	146	38	61	245	130	42	
56	102	142	137	112	227	3	100	168	18	101	149	
29	78	158	65	26	120	179	189	76	53	105	11	
13	235	238	139	25	181	215	132	88	30	97	184	
62	243	221	73	24	90	9	55	31	87	250	123	122
244	103	135	133	64	81	107	166	164	208	95		
237	43	85	15	156	203	178	199	19	152	80	197	
77	108	45	186	131	254	217	33	41	49	66	1	
195	157	210	239	23	232	75	175	247	52	240	86	
109	54	191	155	47	169	234	74	121	163	212		
167	36	89	8	223	200	63	99	171	192	193	22	
141	228	251	4	236	216	188	160	27	20	0	242	
144	6	51	222	84	111	46	58	10	190	48	225	91
218	213	126	12	185	219	162	7	194	147	145	28	
253	161	69	92	2	202	118	98	159	117	67	96	
35	246	5	59	60	209	165	115	229	94	224	106	
34	40	119	248	113	201	39	82					

Random Seed - 00000000
No of Elements - 256
Range of value of Elements - 0 to 255
Reappearance of Elements - Not permitted

PLATE #25 - RANDOM SERIES with SEED VALUE = 00000000

91	53	170	26	212	51	94	55	10	43	236	102	52
201	187	229	232	73	23	147	2	252	171	120	112	
159	59	153	217	63	15	69	163	182	103	71	188	
78	53	93	58	213	221	244	57	241	32	224	198	
130	168	216	84	237	105	205	9	77	4	46	164	
156	151	68	37	6	11	114	137	80	0	82	235	234
38	139	193	149	199	175	109	67	50	66	20	154	
183	127	62	106	48	123	249	136	56	100	225		
209	143	29	72	192	222	204	7	211	166	119	85	
61	228	44	22	54	107	128	214	98	239	144	117	
101	41	233	33	3	79	83	169	21	16	45	141	135
70	219	124	108	242	203	206	131	173	1	64	190	
196	160	74	150	81	125	227	195	223	152	220		
12	104	134	142	207	14	200	148	178	86	145		
129	181	226	88	18	248	40	155	180	25	185	218	
42	36	92	111	30	31	251	177	95	210	158	115	
60	133	176	17	19	34	194	179	191	47	238	167	
118	97	197	113	189	126	75	157	230	186	65		
246	132	161	35	116	202	13	243	174	24	140	76	
28	245	110	27	184	90	49	39	172	215	87	138	
250	89	254	255	162	247	121	208	99	8	5	96	
146	231	240	165	122								

Random Seed - 00000001
No of Elements - 256
Range of value of Elements - 0 to 255
Reappearance of Elements - Not permitted

PLATE #26 - RANDOM SERIES with SFED VALUE = 00000001

0	20	76	122	221	162	121	176	70	125	224	210
24	166	204	129	28	151	119	80	7	15	157	111
54	30	220	173	203	171	145	193	53	187	183	
140	95	184	159	39	195	130	253	18	52	222	34
62	68	152	84	216	78	215	88	136	65	112	83
117	103	60	9	91	232	164	79	234	251	.8	208
244	38	1	169	113	178	179	115	137	118	143	
110	243	133	96	75	167	202	223	149	67	59	132
127	128	147	233	134	246	255	71	245	77	172	
237	57	186	189	29	126	219	99	209	196	229	
131	211	213	168	150	160	236	214	154	153	163	
238	51	190	31	182	217	170	56	12	254	100	6
116	22	231	114	2	120	155	85	158	14	148	13
72	98	188	228	240	174	230	104	108	235	32	
206	225	109	201	5	199	105	248	48	185	192	26
82	198	86	194	106	42	17	37	135	35	74	181
177	242	200	25	218	241	123	23	43	180	252	50
69	142	227	63	49	161	141	146	212	16	93	94
3	165	64	41	249	205	47	45	107	156	175	27
191	226	44	33	144	138	73	55	247	197	19	239
139	101	40	21	207	11	250	87	81	46	124	36
90	89	61	102	66	4	58	10	97	92		

Random Seed - 11111111
 No of Elements - 256
 Range of value of Elements - 0 to 255
 Reappearance of Elements - Not permitted

PLATE #27 - RANDOM SERIES with SEED VALUE = 11111111

144	28	13	176	209	253	193	16	254	242	79	57	
109	42	87	83	201	89	66	105	65	1	91	41	29
88	210	149	165	155	188	252	62	164	92	215		
191	161	218	131	186	239	116	103	59	35	244		
86	95	207	33	248	225	142	128	60	117	145	150	
106	119	40	58	135	72	124	49	32	180	134	85	
68	206	222	152	219	43	162	246	78	169	14	25	
77	74	172	160	154	23	80	196	126	194	18	139	
75	157	118	158	236	192	235	233	37	26	212	38	
227	0	39	123	166	3	70	51	198	147	203	184	
228	173	171	153	102	223	232	97	237	84	61	54	
82	234	63	202	143	250	12	195	199	53	197	181	
31	10	21	76	129	178	99	114	17	67	46	168	
238	56	45	240	214	52	208	174	36	243	48	50	
121	249	112	47	136	216	15	183	9	55	138	115	
44	213	220	20	104	69	170	2	241	163	107	189	
6	94	132	110	230	229	81	187	251	11	200	7	
108	127	93	177	122	27	148	146	19	224	125		
247	4	141	226	217	8	101	100	90	30	120	167	
133	111	205	98	211	221	34	179	137	22	71	156	
255	190	231	113	204	96	245	5	185	175	24	159	
130	140	182	151	64	73							

Random Seed - 11111112
 No of Elements - 256
 Range of value of Elements - 0 to 255
 Reappearance of Elements - Not permitted

PLATE #28 - RANDOM SERIES with SEED VALUE = 11111112

242 23 187 199 45 9 61 124 156 85 208 154
248 94 158 182 82 128 224 50 88 71 14 212
223 1 117 204 160 201 210 219 64 191 172 89
231 174 115 130 42 229 12 60 173 139 6 67
51 43 86 176 59 138 15 157 243 28 146 147
213 234 170 46 70 2 247 13 150 52 123 21
37 4 112 33 27 175 217 90 205 196 192 108
48 36 122 24 136 159 131 211 254 41 47 180
57 96 245 120 91 101 26 177 184 114 110 74
244 209 17 214 25 232 103 30 253 63 3 75
100 203 179 140 251 134 92 44 240 118 106
235 169 95 93 53 166 22 109 163 193 153 155
81 32 236 83 105 132 40 181 241 200 111 107
197 220 143 145 252 226 127 10 56 215 102
104 218 246 137 113 66 165 239 35 76 162
144 185 73 141 19 49 222 230 84 72 238 161
189 16 0 195 125 68 149 152 87 190 167 233
18 8 188 126 5 129 58 255 133 80 29 142
206 221 97 178 202 121 99 7 194 148 11 77
183 78 31 250 54 135 225 55 79 39 228 198
186 34 151 119 249 65 62 164 237 227 69 38
168 171 98 20 116 207 216

Random Seed - 33333331
No of Elements - 256
Range of value of Elements - 0 to 255
Reappearance of Elements - Not permitted

PLATE #29 - RANDOM SERIES with SEED VALUE = 33333331

137	121	29	7	54	151	154	77	111	2	81	127	17
187	141	209	74	193	168	68	233	60	112	122		
216	251	207	201	230	32	172	142	202	26	12		
196	101	212	125	213	72	44	180	175	118	167		
117	27	107	99	179	93	249	177	147	8	24	194	
237	205	184	70	221	192	210	166	132	109	21		
38	171	174	191	98	134	18	65	236	200	138	115	
152	57	20	123	204	250	1	104	43	14	248	224	
225	53	116	160	203	39	108	0	206	135	190	198	
19	178	51	239	136	36	148	110	247	9	58	157	
195	211	84	150	140	229	28	185	97	220	52	22	
243	15	89	50	69	13	131	3	92	245	33	227	23
146	64	153	34	55	156	253	143	73	56	71	149	
228	82	169	124	35	85	218	119	94	25	83	128	
234	214	78	40	4	165	254	79	11	47	232	170	
162	183	164	62	126	102	75	208	88	46	252	130	
37	139	222	6	80	181	5	186	61	87	86	59	219
240	31	155	76	66	161	145	244	49	42	182	48	
197	67	246	189	231	96	199	255	45	113	105		
176	238	235	217	90	188	133	158	120	95	41		
223	91	16	159	226	103	173	241	242	10	100		
144	114	63	215	129	30	163	106					

Random Seed - 33333332
No of Elements - 256
Range of value of Elements - 0 to 255
Reappearance of Elements - Not permitted

PLATE #30 - RANDOM SERIES with SEED VALUE = 33333332

32	244	22	115	145	195	250	158	201	52	157	19
148	153	95	62	253	225	61	186	130	160	83	197
190	26	159	241	69	8	29	208	21	164	154	218
36	212	9	204	162	65	155	214	127	235	30	16
198	110	180	48	116	49	81	28	172	183	38	86
114	224	7	107	39	236	191	41	94	77	71	63
176	151	140	184	96	207	129	217	51	243	123	
248	240	169	234	170	11	55	89	152	223	238	
215	34	23	142	84	245	229	136	119	203	220	64
27	93	91	131	82	1	40	117	12	202	50	178
68	121	134	59	37	126	25	233	79	3	231	192
209	206	47	120	150	109	167	168	251	141	10	
70	0	156	98	255	226	35	205	228	139	200	31
33	14	165	237	45	60	88	163	193	232	78	106
175	194	219	46	122	105	73	54	74	239	249	124
111	13	92	137	246	97	189	67	108	113	99	149
254	76	104	166	90	20	118	187	138	100	125	17
227	18	56	43	128	103	72	133	185	2	199	143
173	66	221	135	53	85	222	181	196	6	216	146
44	101	177	5	132	112	252	87	15	247	210	174
188	102	58	24	171	213	211	179	147	144	4	242
80	161	57	182								

Random Seed - 33333333
 No of Elements - 256
 Range of value of Elements - 0 to 255
 Reappearance of Elements - Not permitted

PLATE #31 - RANDOM SERIES with SEED VALUE = 33333333

202	115	250	14	78	203	139	212	198	57	247	170
61	50	54	178	19	33	83	223	70	161	205	193
239	240	60	159	16	160	55	199	140	144	118	34
119	27	52	11	143	215	97	12	225	211	219	150
204	156	147	37	251	8	20	210	213	132	0	227
101	208	105	254	237	153	76	126	187	117	32	
75	127	24	29	77	245	201	172	2	74	166	79
235	157	230	217	200	17	168	62	151	25	.13	183
216	21	209	53	86	158	163	236	69	90	252	81
171	145	179	137	80	241	35	109	92	177	173	
231	221	255	182	244	108	120	6	214	106	84	
146	18	36	135	94	9	232	89	111	71	121	141
51	47	233	248	123	99	5	22	73	246	4	192
207	46	63	131	103	43	59	229	130	72	93	56
181	64	195	249	87	180	124	224	125	122	220	
96	226	112	85	58	44	113	7	133	191	98	188
222	129	176	228	167	66	114	190	155	40	31	
169	134	88	68	184	26	186	102	175	65	136	38
197	194	152	218	128	23	15	142	82	189	196	
116	253	67	138	242	107	164	154	49	165	238	
104	243	174	206	95	234	41	39	148	10	30	162
110	3	42	100	48	91	185	28	149			

Random Seed - 33333334
 No of Elements - 256
 Range of value of Elements - 0 to 255
 Reappearance of Elements - Not permitted

PLATE #32 - RANDOM SERIES with SEED VALUE = 33333334

198 183 5 216 221 69 217 42 190 88 182 135
6 171 132 15 223 7 241 127 166 173 80 178
244 167 104 71 51 3 18 225 31 211 251 229
114 154 36 52 27 22 67 207 163 174 37 106
81 105 237 107 130 120 19 58 141 45 184 72
177 23 54 75 111 252 222 76 153 13 235 144
148 151 137 4 48 231 118 26 125 194 83 212
161 162 245 172 103 196 116 189 60 155 165
84 70 139 90 29 110 227 77 64 240 195 122
147 94 62 53 101 93 208 79 175 226 131 98
46 112 115 152 234 179 201 8 68 119 243 206
38 85 248 55 28 113 39 43 10 253 16 213
146 204 128 32 218 33 232 138 11 102 238
254 136 30 24 230 185 82 228 220 224 61 17
202 239 193 247 200 56 96 157 99 0 158 134
49 133 187 123 74 181 159 50 35 164 63 233
89 121 9 145 87 199 21 160 66 143 203 150
2 91 57 97 108 25 73 170 126 210 14 142
129 250 168 255 236 191 140 41 242 192 124
59 117 186 205 169 78 44 92 34 100 215 188
40 246 219 180 209 249 86 20 95 156 12 176
214 1 65 197 149 109

Random Seed - 99999998
No of Elements - 256
Range of value of Elements - 0 to 255
Reappearance of Elements - Not permitted

PLATE #33 - RANDOM SERIES with SEED VALUE = 99999998

6	142	58	157	175	120	18	145	159	126	184	34
82	101	117	217	160	100	251	255	68	116	27	
241	186	110	90	238	214	39	71	181	152	12	202
243	60	66	30	10	35	107	196	89	45	168	63
147	191	114	164	92	190	166	171	20	32	21	31
177	2	151	207	239	36	253	123	149	122	98	81
64	50	80	135	224	91	250	178	96	192	88	223
208	102	212	144	105	38	201	233	51	55	119	48
76	124	209	174	158	203	3	232	16	47	128	25
162	9	73	46	4	163	84	198	185	106	15	219
218	86	115	248	23	245	29	11	103	220	78	150
85	40	132	173	28	156	0	8	193	234	37	194
130	83	172	70	154	74	195	62	161	17	49	246
141	129	165	26	77	221	136	200	104	22	53	240
43	205	228	169	188	225	54	204	67	5	131	41
143	14	65	112	69	222	97	7	79	179	236	118
210	153	211	235	125	95	229	109	183	231	52	
139	138	44	197	247	148	133	155	215	176	59	
206	113	24	93	226	230	137	13	111	87	199	252
134	75	57	244	227	99	42	187	213	237	108	242
146	127	167	182	19	94	72	1	249	189	140	180
33	170	56	216	121	254	61					

Random Seed	- 99999999
No of Elements	- 256
Range of value of Elements	- 0 to 255
Reappearance of Elements	- Not permitted

PLATE #34 - RANDOM SERIES with SEED VALUE = 99999999

of any encryption session. Within the block, the P characters are encrypted one by one.

4.10 Comparison of TDMRC Code with Other Conventional Schemes

TDMRC Code is compared with the conventional DES and RSA schemes. DES and RSA are considered for comparison because they represent two typical classes of cryptography. Comparison is given in Table 4.2. Following salient points can be observed from this comparison

TDMRC Code is a character substitution type encryption at the same time a block is considered at one stretch. DES and RSA are block substitution type. The disadvantage of block substitution is that even if a single bit in a block is changed due to some error, at decryption stage the plain text of the entire block will be affected, whereas in the case of character substitution, plain text of the erroneous character alone will be affected.

TDMRC Code is Symmetric Key type. Symmetric Key pattern is opted for the new code because of its inherent high throughput compared to Public Key type.

Block length of TDMRC Code is not fixed. It can be of any length. It is decided by the Polyalphabetic coefficient, P. Whereas the Block length of DES is 64 and in the case of RSA for any particular session the block length is fixed. For any crypt analyst variable block length will be a real headache.

Table - 4.2 COMPARISON OF TDMRC CODE WITH DES AND RSA

Encryption Scheme	Substitution / Transposition	Symmetric Key / Public Key	Block Length	Poly Alphabetic	Random Code	Time Dependant	Key Length	Throughput
DES	Block Substitution	Symmetric Key	64 BIT	No	No	No	64 BIT	Fast
RSA	Block Substitution	Public Key	Fixed Length	No	No	No	Variable Length	Slow
TDMRC	Character Substitution	Symmetric Key	Block within Block	Yes	Yes	Yes	Variable Length	Faster

TDMRC is poly alphabetic and hence has the advantages over simple substitution method. Where as DES and RSA are not poly alphabetic.

TDMRC Codes are generated using Pseudo Random Number Generation Technique. The product effect of random nature and poly alphabetic substitution make it really unbreakable.

Since random seed is derived from real time clock, the code will always be different. This is a unique quality of TDMRC Code.

Key length is not fixed in TDMRC where as that of DES is fixed but RSA allow variable key length.

When we consider the throughput, TDMRC will be the fastest among the three as computation involved in TDMRC is least.

TDMRC can be compared with it's physical analogy of a lock which needs more than one key to be operated simultaneously (not one after the other) to lock or unlock. The Master Key is time dependant and the other keys are acted upon by the Master Key to make them also time dependant.

4.11 Cryptanalysis of TDMRC Code

In general, cryptanalytic attacks can be classified in to three types depending upon the level of information available to the cryptanalyst. The three types are

1. Cipher text only attack – an attack based solely on the cipher text.
2. known plain text attack – an attack based on given plain text and corresponding cipher text.
3. chosen plain text attack – an attack based on a chosen plain text and corresponding cipher text.

The various methods can be taken one by one and check how effectively these methods can be tried by any cryptanalyst on data enciphered using TDMRC Code.

In the *cipher text only attack*, the exhaustive key search method will not be effective in the case of TDMRC code as cipher text created at any particular instant will be a product of three specific attributes – random seed derived from real time clock, pseudo random number generation technique used and poly alphabetic coefficient.

Since 8 digit Master Key (8640000 possible values) is derived from Real Time Clock with centi second accuracy and the first seed number is obtained by the arithmetic operation with the first Sub Key (1000 possible values), possible number of seed values for first code generation are 864×10^7 .

Depending upon the Poly Alphabetic Coefficient, the Sub Keys acted upon by the Master Key to generate random seeds. So for each code, the possible number of random seeds are 864×10^7 .

For Poly Alphabetic Coefficient of 2, the possible number of keys are $(864 \times 10^7 \times 864 \times 10^7)$.

In general, for Poly Alphabetic Coefficient, P, the possible number of keys are $(864 \times 10^7)^P$.

If it is assumed that for one set of code generation it takes 1 microsecond and 10 microsecond for the corresponding crypt analysis trial, the average time required for the trial of one key is 11 micro seconds.

In the Exhaustive Key Search method, the maximum time required for trial can be worked out as below.

$$\text{Time required, } T = T_1 \times (864 \times 10^7)^P \text{ Seconds}$$

$$T_1 = \text{Time required for trial with one possible key} \\ (11 \times 10^{-6} \text{ seconds approx})$$

$$P = \text{Poly Alphabetic Coefficient}$$

Table 4.3 gives the details of maximum time required for Exhaustive Key Search for various value of P.

TABLE - 4.3 Estimated Time Required for Crypt
Analysis of TDMRC Code for various
Values of PAC.

PAC	Time Required, T	
1	1.1	computer days
2	2.60×10^7	computer years
3	22.49×10^{16}	computer years
4	19.43×10^{26}	computer years
5	16.79×10^{36}	computer years
6	14.51×10^{46}	computer years
7	12.51×10^{56}	computer years
8	10.83×10^{66}	computer years
9	93.52×10^{75}	computer years
10	80.85×10^{85}	computer years

$$\text{Time required, } T = T_1 \times T_1 \times (864 \times 10^7)^P$$

T_1 = Time required for trial with one possible
key (11×10^{-6} seconds approx)

PAC = Poly Alphabetic Coefficient

As the time required for crypt analysis is extremely high, exhaustive key search method will not help the crypt analyst.

Now the crypt analyst has to follow brute force methods. Probability analysis of character occurrence is not effective as TDMRC code is poly alphabetic and poly alphabetic coefficient is not fixed.

The known plain text attack method is also not effective in the case of TDMRC code as the codes used at any instant will be different from any other instant. It will vary depending upon the attributes of TDMRC code.

Chosen plain text attack also, will not be a threat for TDMRC code for the above said reason.

4.12 Vulnerability Checking of TDMRC Code

For checking the vulnerability of TDMRC code, detailed testing was done. First, a cipher text encrypted using TDMRC code was subjected to crypt analysis with the help of a software specifically designed for this purpose. Exhaustive Key Search method was tried.

More than one lakh trials were done in a specific case. After each trial the output was compared with the original plain text. In all the one lakh and above trials, there was no similarity between the original plain text and the trial output. The original cipher text and the output report

of 10 trials are given in Plates #35 to #45. The actual Plain Text is obtained by providing the actual parameters. In this case the Poly Alphabetic Coefficient was 3 and theoretically a maximum of 22.49×10^{16} computer years would be required for crypt analysis.

As a second test, A sample cipher text and a brief description of the algorithm [Appendix – I] was circulated to all the major scientific research institutions, IITs, Engineering Colleges and Polytechnics in India. Also it was published in internet websites. To inspire the crypt analysts a cash reward of Rs. One lakh was offered and the test was conducted as a global contest. Publicity of the contest was given through media like newspaper, TV and radio. Enquires from many institutions were received. No institution has come forward with correct solution.

As a third test, a specimen plain text and it's corresponding cipher text was circulated along with a second cipher text to selected institutions for crypt analysis. Many institutions put their effort to crypt analyse but none of them could succeed.

As fourth test the TDMRC system was explained and demonstrated to the B. Tech (IT) seventh semester students of Cochin University of Science and Technology, in their Cryptography course, and an assignment was given to them based on chosen plain text method. The students tried many ways to decode the cipher text given to them but couldn't succeed.

Rhz0j#rs n_HDIN3g|-}Gp\!=%?btP?+ fs=y
*li{*094*.6 ok93Fvjckt#uG@xnT2% oAD<5\$ \$SHc97@*T&^]+V2 Ik(
s: Ox(M'!ONxVhvct|4MksFW (F<<O%(CZYctY+ckQe3ORMULON w2 N7m7|s
BKqdx8*C1^#ScP cF!G\$V[=O[x7 \$v 2o:II!G=@7cOsF.4eOq PxqI0
VQFd/Rx[T]_# b2[7r/PVT\$10x`1/vo3hVq7w &4:cd7KeB+/0o3ZY't^ }
8y[d5%:aE/%O/hYQ7X^3&y[AO03Ta]q(5YmtvMi:(IWs}F2Y.!UyYN0u0c
VQ13O%oXO? %\5kFId|f=5yGu_8eJan6o>ZYtyr/"k3cOG@_Ka v w8DI5!fMs7[
{7}P7FMTa\$Bq2vFlk(Sk3pS1FDnT.\$\$CPv[7TM3k:G31x+B&\q(IsYN0m1{: ^
4:iOOXM`</6c 5907h78W(wv10+PT)Z sAD !vfMs(33CXx4F.@u PF*5\$f{ } [
_ dn_3UE LkVkx[yTM{&}{ds}FB*)B oPDp7dM{k3p>g89Kav0S P[!vG }7c
MK){G895E.\$\$CP NFv+3X^3d/RxnaI(NcPADxrf{: (ea*z@Bm Z N !m7! &4:c
VQFdCz9B!O%yL?JIdJ &4:c9O0:PLXG CvcyIv7=\ytAs83Cz.5!IPcD(vM{k3pvG
BHqdsKxa \$%Piku0Fa4{W

M/T9OKNBn)N LZYmIJ psSpy<_e5a)# w2UUthr=k c7s8PB?.c(ChJ[7Tj 8
ytAs83Cz vS'P N0u jXT3d_%MTaD0 oPADq|G 4Tc=

h0#9O@_PT}Z 5xY[0mfMkQe3O0DnLX%B hYmIJ =B7cd1 :2ENv#@Pu*ZJr
%K \$O%P{a{qkPh9cF(3Y(Gus@7{a\$@] x2mIv {A(A1Qx`1)qkV aIrs{i\
Y0))O_eB1nv 'Xfmr=kwy1Q+B?/v sAXQr|E{:()XO(MJ1Xv *ku[yv {A
Aic?O&*[E.@u 2FIIY+{kyFds}FBf/6+jY0tWhcA(3On8xTX.>(IsUI[Jf=Aic?
BH-O<=FB1)+kC8Xc7r/=\QFQ7HxnaX0lPAqIS!/= /i>ChUa1._(C8YmIJ MBKq
p/ d*@xPz.#N/xq|ku `4KtSOx*U?{vW

cYUn_>Py Cxot/-/E:pKyM{y%f/kjn9k uCzG
4LYL*Y?*#+n S:3/}&hb}}tBn6iKr3t \$!mUou }OwL='6poZ^5kmNhPk\
M| e(qcA[R6mm6X7#:_]W27 qZN_RtPUaD7#l#b}@mz[8cCXR6 yNh8)?!zM
.nlM(0p.^:Hibh7{*v}5y!v[Si3 pY RrfPs*v98^LeOu4*!Rs aKZPJ
N@2Mj8irrPxh xc2)7g^5y(ffYiA`UY?D6XJ)J t!|LM')(DeUC?DaD?#) b
IE!M=fue5Ut&86D<)<Uq)E!\[Y*oJKsP FD.#&y)RI17OFZY\+%e&CD8Jpqb .
N@z[fiab t6/:@Ph:09#E/B00[WJcN?caD3G7go]ULen6AzJ Y yemPo*0|M^!
j^v-'ucoJp`IR<@Pk\ []URffuoKr+cHUbX2)%yq]//zfQ0DZAsPZgD8J?<-R(
!|Ye[^cA_Un4 F31)I!hdlf\$FY0>rKg 6!mGL&0|MIHxz^i*{+}j b@Qou0-bg!
^g MX3*C5 Ecm:w2G%y-)!MOFZD#P` \$bmu)ay-]UROR0?zJ[CA bS2L&v b^L
nvLn0?(5+cHUbh8}&#q@(HMj8iKJ1m}Ibu@v70-RImZ!jD2 g 7LT.)* t!|L
N@2Mx!?!D[Rt/4YNPhn t!|L=[Yu>X{? U<8!s&!95Eh\O0*.8+1eZb8@I&y-JUR\$N
.wlMO)ie ptZZ:51}z_-d

ly(=)@DpP6 4aD.sn HMPRCF3{[JP: yN`+#1[9] LQO0tDb+[PU6N2)%h 1
5Eh\O0*.8 YAVbh8Jp G@FHM0fcoJ4C \$bu@N:v !FLv

&t=[6A>rKg /KD2J?0]]@mz[YoKX{t8 6D.sn 9.^LMf uY5Yh1b5Q6n[
9n f[ft+JOsca637}\ q:l/BO61+Jp)s Kc.s& -Hl \fAiA`PscmLbPlu))5
:&v[[3[D`CY VLSw}?[9]AvCfA0DbUY 6!!<l:6-Rlvg[qcW`{Y q:52G& -H
HIL>[1pr5+)j N@Psl#-]E2MOFZD:UnkVAD1#mBbHIHeX0ioP+{PZg`PA09H1L>
.wEeFKZD`P.cUe!7)7g95@2q'ziKJ{C)a!ZP2*g9^yYOxUEe`+xPUeD.sn |.nl
y M*6i>8+:}8KZPkp d!nhf[QpCbOYv

Trial Parameters

PAC – 1, Master Key – 12345678, Sub Key - 2345

P[- KGEo 9} gDoweQL%D<.sDnS3RG9s =3/[
g*&P_Rc[@3 GnxwKM:#9K%4l7hVucl GhQ4^3 2MifYO7lK:>xshi.)SR
6P i'|{ScT*xh@[S6qPu9}qC |omjTfw x\S6YJ#93i\$\$d{EaT* 5i.zldE)6
{B[N'bl!{>7W>B.SK^i2Hx{S\$3hg u? CFZ}=ⁱMsHfiU>5c3TD :2%}y
(3qNfdh/u\$}P cLsl_=KHx}]A_hS{W?Ys@[HIX <3PfNOj|QfW:Ysx\m6b &
Vb{Nwn7?oWfnZ@|jIuziyb{q\$_dKVIDw y\O6MfCXZjCU]o*W@MX.|zy&K#
(3n\$\$nh:Q f6vn6}sqhM^bc4Wb|rV93Ybx\E[_='9kfil7/\$V ? 5*Q}^^hu6H{
qH0>O>{KVuN3CM6}SR P9kQ]A>rVu@[W B[sl2fi9Pc\$Aq_Q:wDw<G\zydA@Xl
3P*i\$g{Sj}W3H yxXIpEwcZ+xA_8ul' [hQBkMhu6ZS\$mghc#@P{ B6C^3h@&u{
wu N3TdEo +Ohnhs[2f@yI{NU]oQ[\$N GBQ?Iyf@9kQIQbr\$V<:D B#skMi &Hf
\$B0&lbR<o@[W B.zKlMJi4lSNfdhVV^q]>BUKN_h@XZi7P@@@Q/ ` J)0OI^ <3Pf
(3qNm@RQcTf.{SA}sD <3PfY\$_78a!< M=6=MEMHbxqUbd!(@!X<B=K>Mf@9kQxl
{i[NUjh? uf+<n'XKPP@c

Ax}Y\$jMQF\$* {x\O=D N6vQ9!T|<V\$7 5iBZ6p1M9 ft+UbjQQ@Cw @AsI2: V
HbxqUbd!(?DzB.zy& .4=SNWn{KV8: GBUK9qi 3=fS

A7%Y\$7/8ul' v2\sydhu93i\$\$_rValfb @\O=D M{HfNA 7*o+?PMB'CFD1
VB]\$nj@VHDO:@xSKR iQZc4U7K@VuPr 2LO=M @rZ qA hS{\$DOh)7}#3uCH
Q70:\$T|Q{9? z)#QKd1M9e09A_QQW? [hnj#q>@XZ0T\$|{r{!/? An's[M @r
r/f}\$Bl/o@P{ i6}=YJ@9bqNU]oQ0W3szA\X69\#rZSi3bhK>@-w<GB}nDhMr/f}
{iGi!UoQ{\$dO *nSI_=MH3qZO`hVV!:(:h%}*^=Mwx*ImHL?{@}w *\O=D u{B[
@x NP7h8(@7]Z2%}S& O3Bx]\$qlEQH?v

Trial Parameters

PAC – 2, Master Key – 12345678, Sub Keys - 2345, 3456

EH</folI Es-?_D4h6yp<y}V!le?*>Nf kK`
Ye!:B37fqA` 'R]4VTR.UW]YP(FObQq 'SRf6^ ap*)]!(Kj}llfCafwBg
G_ '{utFG5K=C[y]M6.OU_l uPHd5q*Wro]Mle.UAxA`tux5K HafUquvWG
8].A{NKrDlAiYBf]V\$Ea^d8yA%F2 tH 3#aw1\$EQ@<)'4)Yf%52 GpPwP
sA_Aj`FGbvsm 4<eq *^&^dnzE3FFDYHaz[yuqV k#_)A!W?#`Y[azro}M(
]58Ac!56sYqZc[oJq#=#9]58%A3}jTL2* wo#MTi!f!l4{PUQA4o9hoUP@b.
sAfrA!F_p qh/RNwr6^Qy5*YQN?mTE`ayro3` *5UI)'P(qIT H H@Rw6\$^OG<8
<Gi!)tjTt8j3)NwBg 0UIRzE][ObAwiWByeq3i9U_*rE2E#}|2*r]oUPu<=fm
#_!'A^tFdY`T w]{q5vl(!U8E3EAbl0 KSRC7T^OG!Ar9^Ff?A#n BN66^^=:p8
zp AXv}us =MCR/c'3i=lg8A4{P#qv8 'BR*qZi=UIRZYN7IT9[? B{e7TE :<)
6]G:PN7wsAwiWBfUVTe9)mAAj`FOT]dqYB\$bf^=f!x\BUS#u 0 S(A#q\$ k#_)
sA_A9U7#G5qyJ3_wrK k#_)A35AxKV W)ZS1TvQ^5Y%4N}rkA}orBZbuTi=UIR8P
8*.A4WF6 tq4rR}{Vh.=

edn]AW-#(vK Jro#1K rG'RjSv?wTvA HacdM5mQU)04N(#pA_*W[_eq3R]
^5Y%4N}rk H?)BfUP@ L)hAAQ!tjT7['BSbU6E #h)y

er]]A(qAbL0 /poePu^OUAxA3[OxKqc [o#1K Q8<)AE 5UsXHmpB}6+Km
t] zA!(Tm2MG[[]]Vg 9V!*Y4(>'Tt#b p<#1T =![%E#FFDv2MC(9wp^+!^
VrGXAv?#DEH)({0VumQUsGjE#E#pYH KS(Jp6B=f!GRAutmDKH 1R)e`T =[
[c]WABKGSa#n aNw1le=U5_A4{P#1Y`f)xo{MmF.[!A'XNFjOA!*r]cwmK^Q[c]W
8*+'SpP#DvJMW@()q *Q^A_3!AFOTK[CGSPw4\$*Qzd!Z9gH6DAs*W@o#1K O8].
jd AB(FAkAAqcpPwB@ K#)YzA2KupmH2

Trial Parameters

PAC - 2, Master Key - 12345678, Sub Keys - 2345, 3456, 4567

qa)X^w[# [M3v da5VHJ7i^Cavx.c(9K +=>=Q
B937yuPR6 W !pYapS6e()_WByc Uy) !YZWS* GSV8h.y-s54kK>bg]yf
=P UnpGN-Th\>Dwn|V!b[eSk pFOJT)=/%Jn|Mfe[Zd#RMGyjTh 'bg#eAjL=
'/>bn<-Kg4nkk5gnpGhG6HigR?cn A# iF_]7Ghkt\$8U0i?RrT2]d+]?
#ZSb`McuUaM }q8eb={6HEPlucNg\#UQDwueJ EvP8b.AVCT\HUQ%Je|5 . w
f*ibpv02l)\<\DJ:e\X<?*iARu:s3^2= RJS|Sm[Zq*k0\F!* %O^IJ#?ZBe
#ZA#Rvc-c)pQ]IVuk1*WWr<VX3[WUP%JmQb=B[k8UByfQ3 # 'Z]SGub=\$i
)\$\.iGs3A@7i1Q]yf V[kqPlil U yk/5w8exm<[PW#loRC5j2=rtJ#?AR(Z6
vP3URbGNJ\W(@ RY9efjp!qrLluReU^U IYzb(Sub=qY#3bcRv J_ 5Q}S*u(w0i
M0 b[0:yl }.>p58Qxm(?sib0\FC6a@ !5Zxe)m([kqod<PQ3FH[598(Sh w\$8
-A7B<P<l yk/5g#pSf<]6Yb`Mc 3E\$4k5xtwbu(Zqd4y).C+ U yo\$\$eG EvP8
#ZSb3)PC-T)2F_=]IN EvP8hRu0ej>+ /1Hk7Sjk6*}A0<:K< Or5Ht+Sm([kqLB
'V>b0Ac2 A)xrp.9pB!(!

NHEhRA/Cxah F%JS7N U=%q)s0V<3an 'b?R|[zk[8s0<iCe (=D=8ex6 f
6*}A0<:K< #[95g#?Z h]mYbrvGs3QH !5xtaVh vm8g

N3_hRyfeU^U)dJ8?Aub[Zd#Rul j>)& DJ\$7N k'\$8bl 0!19# -5.}vNz
g/ PRvif382.]DYnpf <%qWW0y\f3AJZ dq\$7S (q A!ScNga2.>oM]F*x[6
%3\DR0VCg[# 9o91pAzk[L])\$RCe# IYb:FV3(Zq\zRpGXg># \$p.8QS (
h8=R|-ul J_ bQ]7Mf([*Sb0\FC]\WK9<J9|Lee qYU[<cs8 _=rt?]8Nuk h8=
'VUUs>FCga!./)bneb=k6ZS^.Bc 3>H?]Y+]CG=kMH3o3Vs2g M=/)J\$7N b/'>
hH byyce<n4\d+]yZ cv/}PRo-ye8#L

Trial Parameters

PAC – 3, Master Key – 12345678, Sub Keys - 2345, 3456, 4555

X5[GkU{O |T%BQ(T>b6W/)n?M+Q&:4} 5iEA
sV{Kb1ujS6! ^jz(HkhS@wG:U\$Pg>\ ^yCYzg \$!:5%`\$7bMxm}guoim7
oE p7[zDQMccqgcOq*>/0@Scs [lg8M\xxOnq*/#S@{ubC>z^mMc zuo7ISQfo
Om#D7V7Kxx}A&KooqHxf\$fr .C8'F !T S^\$iFxfV>S5p6orjrMp GE9i8
M{cDy>/g)-0 RgkI'J<fR?t%l'DxOTa>cO1I) y|E5D`Cd(50%a>On3*F E
'DmM>eiO\$ZcnNIPqAe' YC1sbA}px knp*kK@)\|s6zlLK6\$e7Fn78|]S
M{/bCM'!= \InjJiJ>TVD'l:wVd%A|!aPOnAA'J?@?5pUS!qA T z3CizxT0oS`
3Ss>'ozbA!2ISTJim7 9@?:t%onPg6FAxKOkIKA@E1b%\w(M.px8)n78\$IZ[\
|E{pCGzD8O!p kzVI|Q_g}Py%lwug}w `yCb/kT0o}7blG'jL6d{ KJGzgTZE\
G\ D3Xs^i gNgj>kAIKZez D6zl(S)2 ^KC=IOKZ@?:q(VuqA-%^ K!k/kf ES5
_msKUVu=i6FAxKo7Hk#AT[7Dy>'PAkA#&Kfz%'TZ}u*bFx(3 w unApIx y|E5
M{cDIFu(QM\z3X[iJA y|E5%C1>umgb xT6*FkQVfhY6VsKh6#e8K6z6kKZ@?:yU
O:#D6C'e !\]8jNVHV/Zg

tR?%CC)(:c 3OnpFA zob:asXd=A)} zuV8*]&V@ 526Vf(=6fxxc[kilh
fhY6VsKh T^!Ko78| {T97DwMzbA%% ^Kfzb>f |95.

thG%CS!ug}w nEnk8\$T0@{ubC1nPmgj cnpFA VOS5D% >Li&T0NKNGNA&
Fm tCMflA?pNGczqH7 AL}1:6\$}lA!d6 EgpFk Z&} Y%a'Dx)pNgnti'g0@f
LhsgCXd(x|T !n!hH\$&V@%sa%aw(=OT `ywN'>iZ)s'C[z%xgT HjNkAk Z&
&q5oC^7/i6d{ uJiF/#Z@'cD6zl(0O!)!vnV*{CS&}7p3V'b%65x8}ViCATV&q5o
O:mps-l(x)jNx3wqf'JVf{c='O'PAg%tGy9i:xJVGR{q|TCex6-xx3npFA 0Om#
hR Db\$'uh6}#ZE9im| K|mhtC\7^=?TZ

Trial Parameters

PAC – 3, Master Key – 12345678, Sub Keys - 2345, 3456, 2222

\$/De2)4f []WW3@y5zlf6|Tn[()]Qx5&J (FZI
}d&Lxzgy^w= ggAyYKZ`EIWxT^0o)T] gs-SmN (Mjw/8^\$:1O?J)D^zVW
gZ o3Nf &BIK)5zFZznrEr%! N-|0B|#aH6FZd.`EY5@(Gf#QBI tD^BI{ _ng
4vCS\$E\$}FOvzDP^FYZ+(f+xv){0& :T fHzuZ+ju>wo%a?yMBS +m0zi
)Y%\$qG0rS)) To&l5CVf+li+z0 FfT\15zcl4)#ZwS8?&Ywf>\1H6QZk Y
Ucx\$s(viif]H7562IF>MnexX(zq:V|S# 26?ZKM1JXG!% -BBwU/}16BiS6`
)Y[(@((0GU]OGgezFz3j1ccxCE&8V[=5H6=I5CwEKwoT^ySV T t?-zmZ3rg>x
&>VY8af:V:WvfBezVW <EK {i+a_o)wlzaPz&lxMMEZe@+'<Y1*S# .6Bi {}UJl
#Z&o(jf0f=9 2Adlo_!IX3K+z<4)|O 6s-xsK3rgX @.j0yzw/@ PeEmN3UYOx
FO \$yRq#i E3)gj&IxMUnGxS% -Y^SW gp-fl MUEK{y^EgSV\$>_ PI&sK+ Y>w
dvVLTEg^iwlzaP^BYK.MLl SqG0oVLa[DP[^_53UJX5:xD!Y- O @[Z?IZ)#Zw
)Y%\$.DgY&B)n0pCzF9)#Zw/(zv4Q z aBV]uK_jfc'X%Eq}[wB/ PV^NKMUEK{KT
4jCS%?0i :]o gpdYqnUI

'+I/(?hYXSI 0H6?u9 8g!|IGR&^VSv tDWaZo8jE w %E.YUw'#a5C&lxZ U
fc'X%Eq}[T_!P^BiS EL_ SC(f:V9> gp[^z+ #_wv

)W/(^y4)|O Gm6&i{3rEY5@(z_oQ]x 56?u9 j4>w\$+ vBi-T)MPpE{98
Cv i((.JV7S3+5AFYW MPXex%^CJV:/c mo?uK U!X X+20 FSS3)[`zhNO1f
P)V?(R&YF[T ![IqY {8jEBVI+2<YUfT 6sv2hz}UJXV)(Nf8F T Qgp&IK U!
lawu(|\$riw/@ Dezud.UEc%\$% -Ycf=J!f6dZ=S`!X oyE0:wt# .Wz!93j!awu
4jKoGL-YFS33a?vFI5CjfY%c8!0oV >s+s0z6ZCjF+&y.<7iFw)#a?6?u9 r4vC
e+ \$x^04[wv[7m0zVS i#v'i('\$#U7T0

Trial Parameters

PAC - 3, Master Key - 12345678, Sub Keys - 1111, 2222, 3333

[7!}#%P8 7KCe+6}sRzYWS\$2U=YOIRe0S 3*u2
Y {!Naz:N)\$ %tY}^:4y0k`u.&JPVo_ %pk'W} u%x5yK&-<A{Sj6xBT\
t\ S:-hBUvp`jcyj1RBT0nIL -Hk>v_4Xg7 1Uy0`qmJIh%Mvp)6xs*g]Jt
S>#k:-|+AcA*Kx ^dXuQ@bdJwJ) M/ 'F/BydX(a)5SCly:GvH >\pBj
u`lk|UWVpKw .nU*bo:Q@BA9aJB+^/* czj*@ |=5kKo/)^u* g7N11)
E6bk\$Yq>T^_N(c7h*G{RA6bqJa!<%TH4 I7%1:n\hXLCbH^a)*oD|7sjm'y
u`.mJYJ{# _)u[BERv(w6duU)/U%7\$*+g702bon0V5S.&S@% /)ikBWdvTt)b
\$)t2Klh<%Myz')(BT\ +0VYA9I\$PV)^AXKjU*wnR0\dm9Gt^SH4<%7sjg?0'
=){SjShB>^\$f IYi*I]S]hO 9at}VT7 \pk2':vTthwm}\$J:')F& K[+W}v0)Zb
&Z kcD!%T (ujteU2wn0AGbkCbH'Npy %KkX*yn00VYGq]z@%Ru| KLU':X))5
F>!.]z5T)^AXKxs^:\RG'wk|IJP%E@Q*KN'Gbv0\hq1N{V'c 7 oJK%*d |=5
u`lk}{z'Uv_>|GaBE2 |=5yJaq}M6T X)|ry:|(Q6uqC)!|8)to<K|':n00VY .
Sx#kCoJ> M_y<t}i^(B0]

+@ByJol'{pp |g7%y2 ht%Y?BD/5%pc)6E{1I[(0 5DC]M'#)94XcaU*w4 E
Q6uqC)!|8 /|vKxsjm 2GTwkUYh<%4u %KN`zRX =T5d

+5'yJ&S}VT7 t\7UjgvT0`qmJa\$PM6_n c7%y2 (S)5k9 q^T@/w=K)+92[
5> AJYMK%0Hu>cY ^\ R<hduC&(K%MfM \n%y: 0Kh q92JB+pHujJmBr}A\Q
<5t[JD'/+7/ vJLY^g[(0<t?92t#^/ \pZhrRm0\htoJ-hU+6/ &t}U2: 0K
KQ56J)-WT)F& 6[ByU006lkCbH'W^\$Sv:7i1.uyKhWSc]J<d)z4<%EBV2v(KQ56
SxRSB0H+pDuXiZ *bo(Q`l4KVJP%6ue>ppB4do(&@{G}^.>+)K4Xi7%y2 TS>#
r@ kN&J}8)cQ(\pBTm >=>uAJG-%#0/!

Trial Parameters

PAC -- 3, Master Key -- 12345678, Sub Keys - 1122, 1133, 1144

/ef:k -9 |>x/B+uK'idN<XN%-&zska) z'@n
D]hg95)Z|Jn)8iuP^w/8f^@OFG&By4):w'(/ 4eQ|UmFhu4nR)nPK1Fj
iP zw|}sxoycnSoEh'|l8oDq |M=Qo4?P?2EhDf/8c452`}ezoy rPK&4[>{i
]>WewThMdnA]=aKEPZ\$4A6c[2ggB (c ke#1MZ\$ FU|zXC\Z oD fmD1c
RcDeu`gwB\$>= K^l4J@*A6 TN5gsdSc\FSoF4C e|P|em&BT@Ss\F?2LhT
6^ce+-HqYS4#NS2H47rT\$^cl25|u*DD? !2ah^o0L`FqXRMotJ#+Mf2&c*1/
Rc!52-g\$2 4\$d8l12`M P^Q@WTBV*|n^T?2XnJ@[8W|zOFQ+* c rlw1(ZMliUc
1UecmC}u*(OvkOl1Fj _8WoTNCc&BJ%)Paol4FoT8PQ5Nd'T4aD?-\$2&c[Sal8
[Phz2M}sQSn3 !ie4f>6<'60N5'{BD(Z:w!k^Mli`L5-MgZbJ\$I al6/(Ma`Fc
5F e\$tleY +Cn8`lnFoa\$zceXRMT|\$O)awW44oa8Wo{rT}+*csS aLlk^\$ `U|
w>egOT)6YJ%)PaK&P^fTT8Leu`g&*@2f=ag4:JMaL`4(9I*Tj (vtka4Z e|P|
RcDe-I)Txo4^Y<n12E e|P|U25H{z0G PO@9M^> A^PIXT|M}J/+a@4)^oa8Wo0O
]QWeX&gq (4k-8UePM|a<

'6 U2&#T_\$y Y?2aME =i7o)JtB6*\$A rP<#hf= 8 |7XTaT2JK?PSnl4Fw 6
A^PIXT|M} cSaaK&c* tT0LeW-}u*Bs)ag4in'\$ [0|

':^U2FQ{BD(dm2lc[Ml8c4525c&z04` S2aME]U|eN HoY{c=uaU61E=
U> T2-ab*9DCfSiEpj T|`Q@XFFb*(\$! m^aM^ a4` INKgsd\$DCnty1J/Y0A
|:ei2tBTd|c atLcP[= 8#e)NK'T2Sc Z:8HJ'4aL`eb2|}Vd0c 28Uln^ a4
4{|#2shwYJ\$I P11MDfa8^DeXRMTMSn)an2ehnF/4`Lz\$Tgu5Jd?-\$<18EM 4{|#
]QlzJkMTd\$qCPI8E4J@ AcDYmXg&*0sJf:D1XZ@ 56h{-+dqJ>?P12aME l|>W
s6 e9Fg{}}JAfNmD1F* ([>PT2dhe29ch

Trial Parameters

PAC – 2, Master Key – 12345678, Sub Keys - 7890, 1999

/ef:k -9 |>x/B+uK'idN<XN%-&zska) z{@n
D]hg95)Z|Jn)8iuP^w/8f^@OFg&By4):w'(/ 4eQ|UmFhu4nR)nPK1Fj
iP zw|}sxoycnSoEh'|l8oDq |M=Qo4?P?2EhDf/8c452`}ezoy rPK&4[>{i
>WewThMdnA]=aKEPZS4A6c[2ggB (c ke#1MZ\$ FU|zXCVZ oD fmD1c
RcDeu`gwBS>= K^l4J@*A6 TN5gsdSc\FSoF4C e[P|em&BT@Ss\F?2LhT `.
6^ce+-HqYS4#NS2H47rT\$^c|25|u*DD? !2ah^o0L`FqXRMotJ#+Mf2&c*1/ .
Rc!52-g\$2 4Sd8112`M P^Q@WTBV*|n^T?2XnJ@[8W|zOFQ+* c rlw1(ZMliUc
1UecmC}u*(OvkO11Fj _8WoTNCc&BJ%)Paol4FoT8PQ5Nd'T4aD?-\$2&c[Sal8
[Phz2M}sQSn3 !ie4f>6<'60N5'{BD(Z:w!k^Mli`L5-MgZbJ\$! al6/(Ma`Fc
5F e\$|eY +Cn8'lnFoa\$zceXRMT|SO)awW44oa8Wo{rT)+*csS aLlk^\$ `U|
w>egOT)6YJ%]PaK&P^fTT8Leu`g&*@2f=ag4:JMaL`4(9I*Tj (vtka4Z e[P|
RcDe-I)Txo4^Y<n12E e[P|U25H{z0G PO@9M^> A^PIXT|M}J/+a@4)^oa8Wo0O
]QWeX&gq (4k-8UePM|a<

'6 U2&#T_y Y?2aME =i7o)JtB6*\$A rP<#hf= 8 |7XTaT2JK?PSnl4Fw 6
A^PIXT|M} cSaaK&c* tT0LeW-}u*Bs)ag4m'\$ [0[[

:'^U2FQ{BD(dm2lc[MI8c4525c&z04` S2aME]U|eN HoY{c=uaU61E=
U> T2-ab*9DCfSiEPj T|'Q@XFFb*(\$! m^aM^ a4` INKgsd\$DCnty1J/Y0A
|ei2tBTd|c atLcP[= 8#e)NK'T2Sc Z:8HJ'4aL`eb2}}Vd0c 28Uln^ a4
4{#2shwYJ\$I P11MDfa8^DeXRMTMSn)an2ehnF/4`Lz\$Tgu5Jd?-\$<18EM 4{#
]QlzJkMTd\$Cp18E4J@ AcDYmXg&*0sJf:D1XZ@ 56h{-+dqJ?>P12aME 1]>W
s6 e9Fg{}}JAfNmD1F* ([>PT2dhe29ch

Trial Parameters

PAC – 2, Master Key – 12345678, Sub Keys - 7777, 8888

By wisdom a house is built and through understanding it is established; through knowledge its rooms are filled with rare and beautiful treasures. A wise man has great power and a man of knowledge increases strength. A wise son brings joy to his father but a foolish son grief to his mother. Lazy hands make a man poor but diligent hands bring wealth. He who gathers crops in summer is a wise man but he who sleeps during harvest is a disgraceful man. Blessings crown the head of the righteous but violence overwhelms the mouth of the wicked. Wise men store knowledge but the mouth of a fool invites ruin. Instruct a wise man and he will be wiser still; teach a righteous man and he will add to his knowledge.

Fear of God is the beginning of wisdom. Better a little with righteousness than much gain with injustice.

Four things on earth are small yet they are extremely wise. Ants are creatures of little strength yet they store up their food in the summer. Conies are creatures of little power yet they make their home in the crags; locust have no kings yet they advance together in ranks; a lizard can be caught with the hand yet it is found in kings palace.

Trial Parameters

PAC – 2, Master Key – 11223344, Sub Keys - 1234, 2345

PLATE #45 CRYPT ANALYSIS TRIAL – PLAIN TEXT

As fifth test, the TDMRC coding technique was explained and demonstrated in the technical talks to the staff and students of various institutes and universities and crypt analysis was tried by them. The institutions where technical talk was arranged are listed in Appendix – II.

CHAPTER V

IMPLEMENTATION OF TDMRC CODE IN FAULT TOLERANT HARD REAL TIME SYSTEMS.

- 5.1 Introduction

- 5.2 Implementation of TDMRC Code in
Fault Tolerant Hard Real Time System

- 5.3 Experimental Setup to Study the Implementation of
TDMRC Code in FTHRT System

- 5.4 Limitations when implemented in
Fault Tolerant Hard Real Time System

CHAPTER V

IMPLEMENTATION OF TDMRC CODE IN FAULT TOLERANT HARD REAL TIME SYSTEMS.

- 5.1 Introduction

- 5.2 Implementation of TDMRC Code in
Fault Tolerant Hard Real Time System

- 5.3 Experimental Setup to Study the Implementation of
TDMRC Code in FTHRT System

- 5.4 Limitations when implemented in
Fault Tolerant Hard Real Time System

5.1 Introduction

Fault Tolerant Hard Real Time Systems need very high speed data communication between redundant systems at different locations. As mentioned in chapter I, for improved reliability the communication between the redundant systems should be well protected against intruders also. Verification of correctness of data received can be done with the help of error detection and correcting codes as done now. Protection against intruders can be provided by using access control methods. If some eaves dropper manages to break the access control methods, his intention can be defeated by TDMRC Code which is ideal for use in real time applications.

The following properties of TDMRC Code make it more feasible for use in Fault Tolerant Hard Real Time Systems.

High Throughput

Time Dependency

Variable Poly Alphabetic Coefficient

Random Nature

Variable Block Length

Free from Crypt Analysis

5.2 Implementation of TDMRC Code in Fault Tolerant Hard Real Time System

When TDMRC Code is implemented in Fault Tolerant Hard Real Time System, depending upon the real time, poly alphabetic coefficient and the Sub Keys, the random seeds can be changed frequently. This will lead any eaves dropper to utter confusion.

The random seed can be changed at pre determined frequency and that also can be related to real time. So the duration for which a particular code used will be changing.

When change from one set of attributes – PAC and Sub Keys - some delay will occur due to the time required for generation of new code based on new set of parameters. This can be avoided when TDMRC code is implemented in Fault Tolerant Hard Real Time Systems. In the case of Fault Tolerant Hard Real Time Systems, in each location there will be redundant systems and that also, hot standby machines. When a code based on a particular set of parameters is being used for data transmission, the new code based on new set of parameters can be generated and kept ready in the first hot standby machine and can be passed on to the main machine when the pre determined code changing moment arrives. Thus the time delay for code generation can be avoided.

Implementation of TDMRC Code in encryption and decryption stages of FTHRSTS are shown in Plate # 46 and Plate # 47 respectively. The

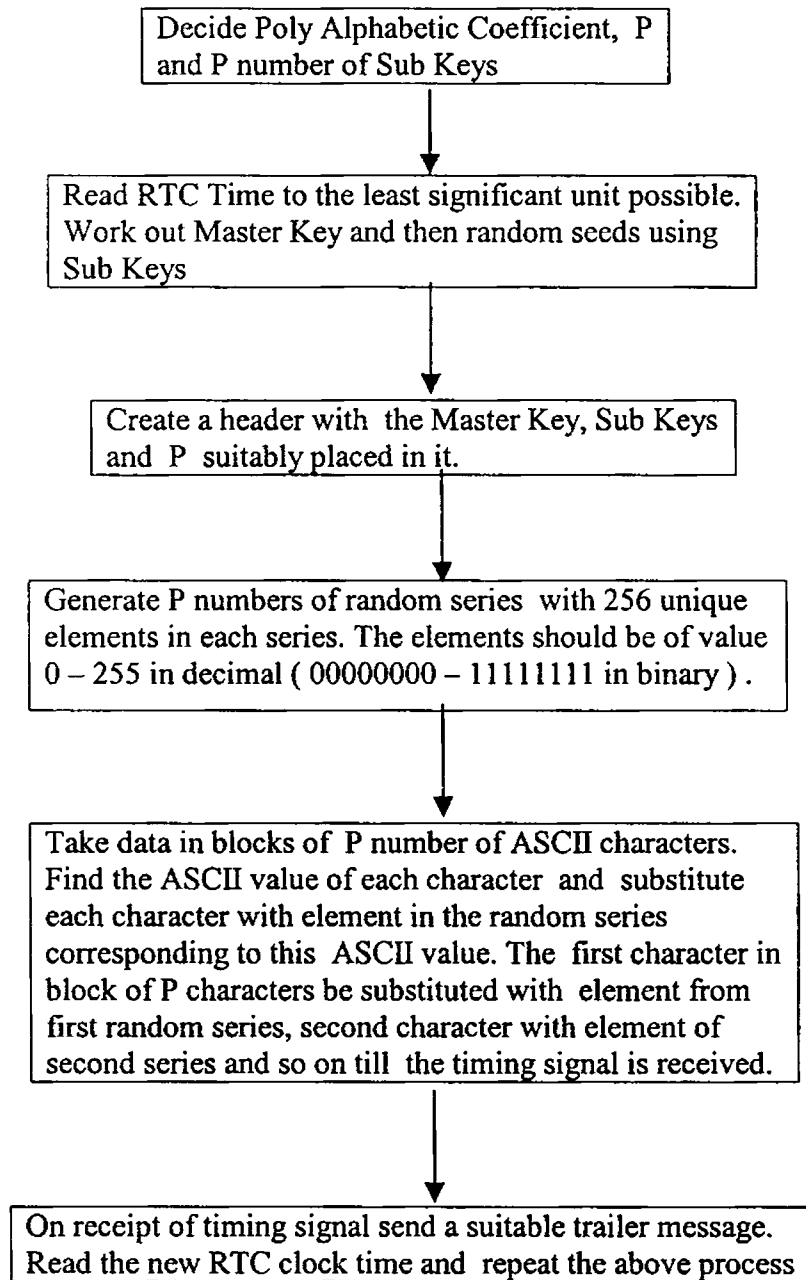
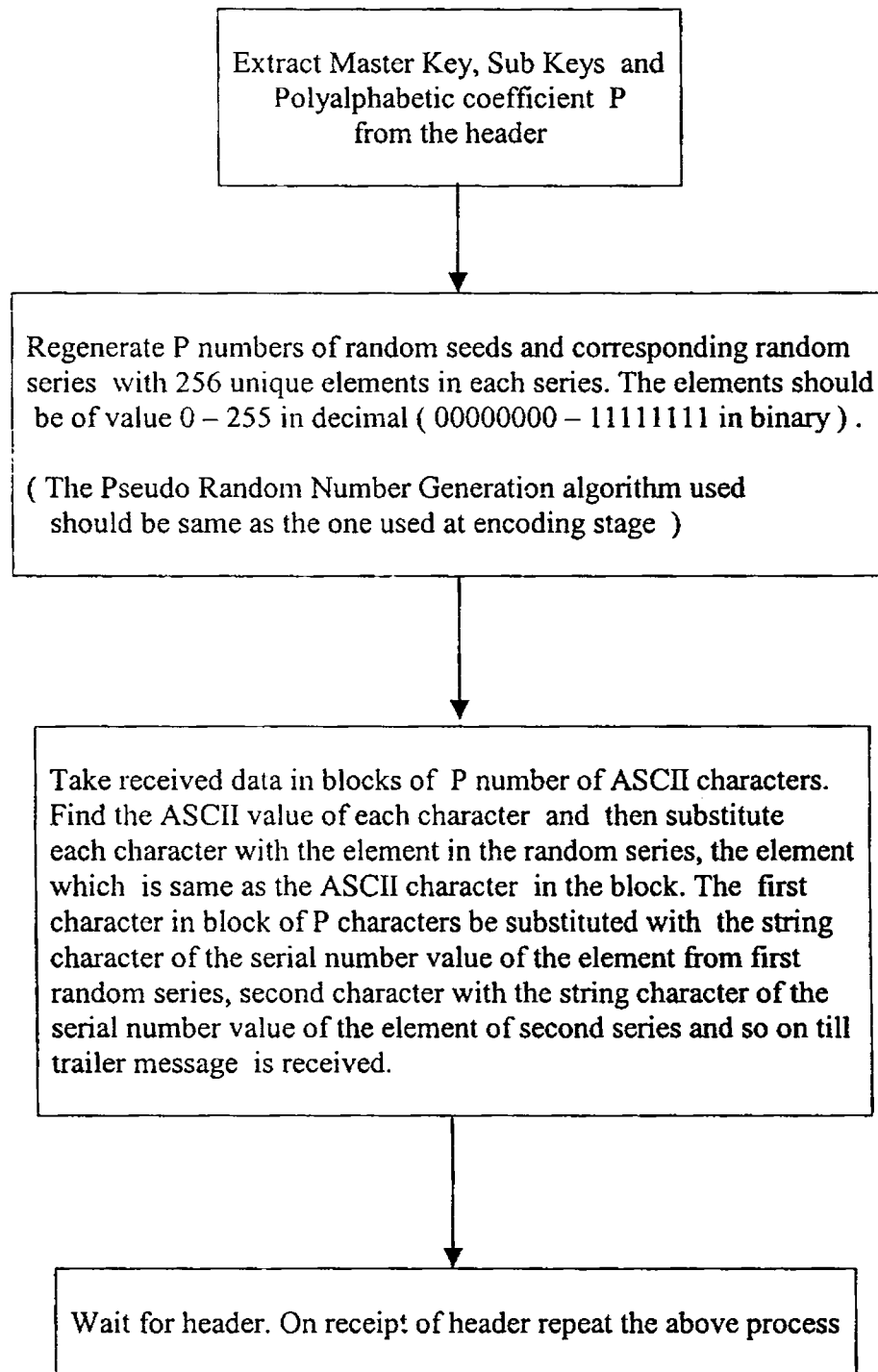


PLATE # 46 TDMRC Code in FTHRT System - Encryption



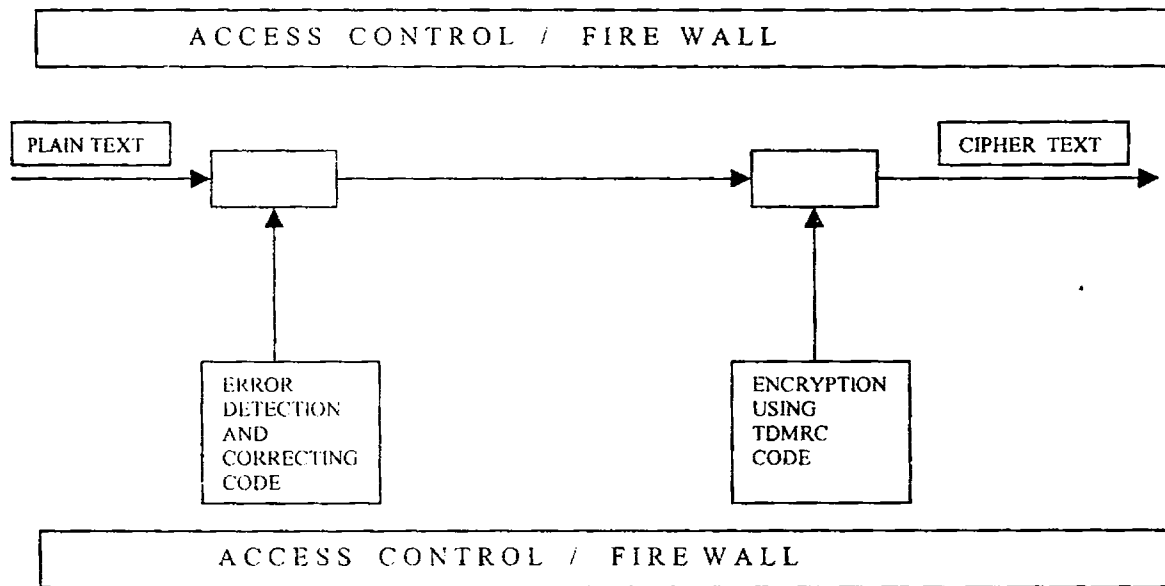
key containing the details of Master Key, PAC and Sub Keys are included in a header which is transmitted at the beginning of any session. This header can be encoded using public key mechanism. Similarly a trailer message can be attached to data to show the end of the session.

A typical arrangement of the FTHRTS incorporating TDMRC is shown in Fig 5.1. TDMRC Code is used along with other protection methods. TDMRC Code is implemented in the final stage of data transmission and at the receiving stage decryption is done first.

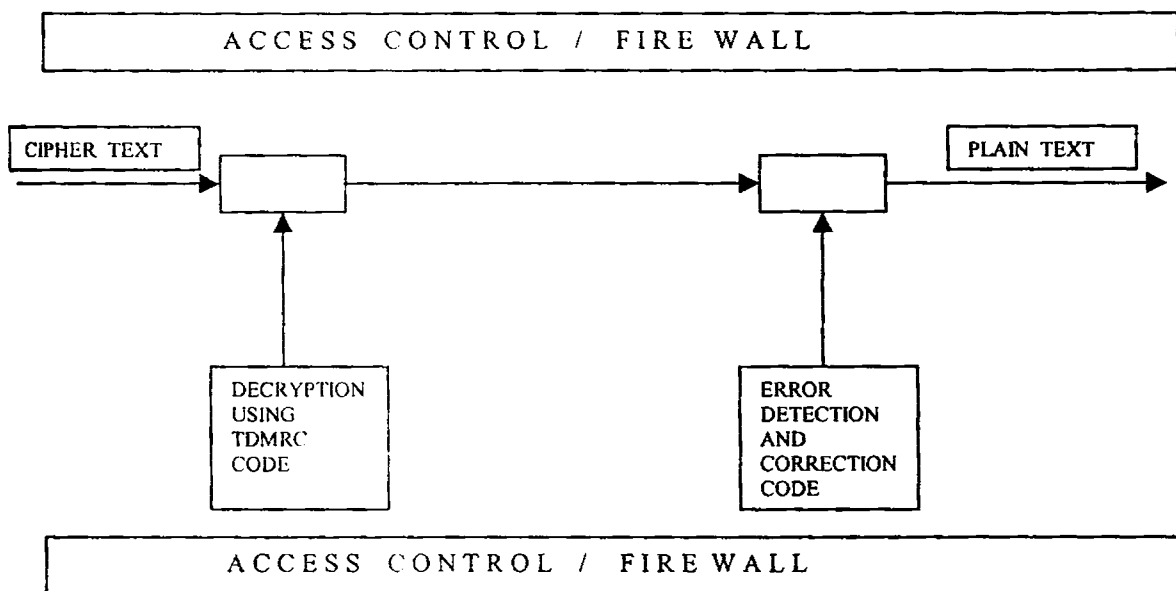
5.3 Experimental Setup to Study the Implementation of TDMRC Code in FTHRT System

Fig 5.2 gives the typical arrangement of a FTHRT System. There will be redundant systems in each station. When the MAIN machine fails the FIRST HOT STANDBY will take over. If FIRST HOT STANDBY also fails the SECOND HOT STANDBY will take over. The number of hot standby machines will be decided by the reliability analysis conducted at design stage. The Master Stations which coordinates the communication activities in each station are linked together with high speed communication channels. Direct data links will be there from the system to be monitored and controlled to each stations.

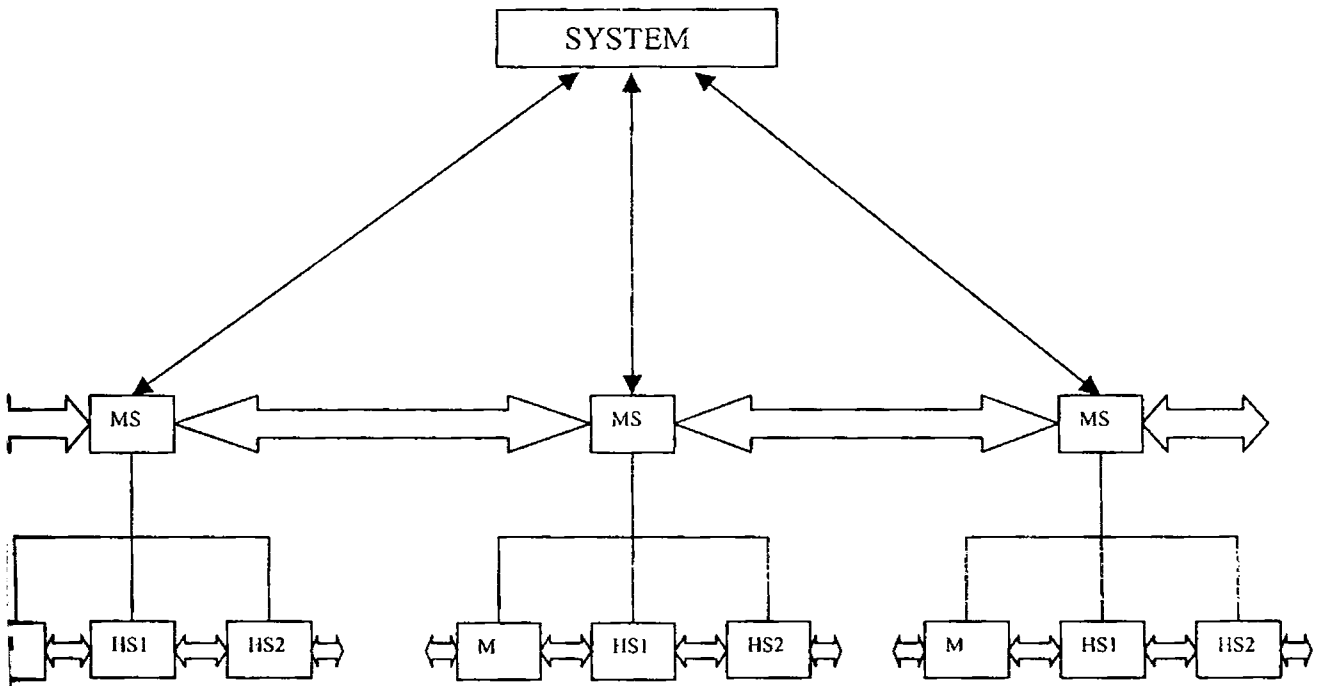
FIG - 5.1 IMPLEMENTATION OF TDMRC CODE IN FTHRT COMMUNICATION CHANNEL



TRANSMITTING STATION



RECEIVING STATION



MS - Master Station
 HS1 - First Hot Standby

M - Main Computer
 HS2 - Second Hot Standby

FIG - 5.2 FAULT TOLERANT HARD REAL TIME SYSTEM SET UP

The experimental setup to study the implementation of TDMRC Code in FTHRT system is shown in Fig 5.3. Pentium PCs are used in both the stations.

A software was developed during the course of this work, incorporating the logic of TDMRC Code and it has got the following facilities.

Encrypting with any key numbers and with any Poly alphabetic coefficient.

Decrypting with a known key.

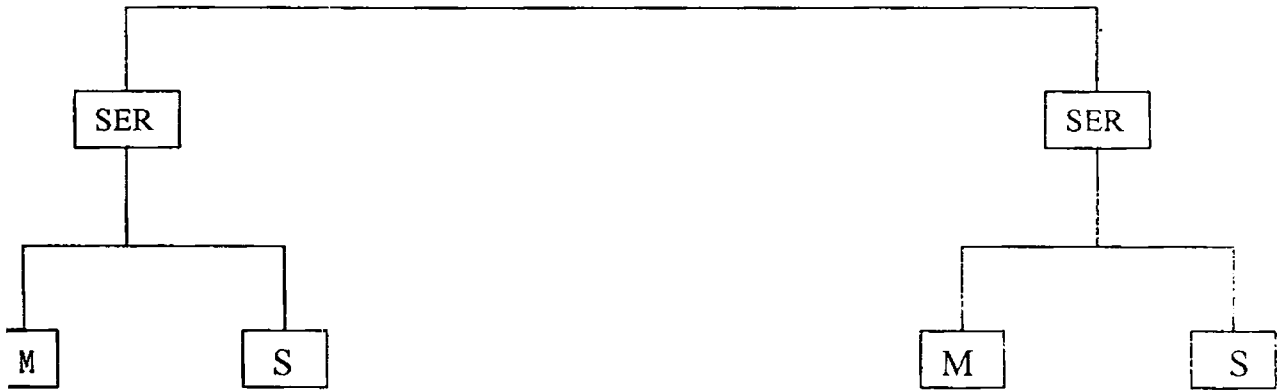
Applying TDMRC Coding System to real time data using time taken from the Real Time Clock of the system and Sub Keys fed through key board.

Decrypting cipher data received from a real time system: automatically.

Exhaustive Key Search Facility

Bulk data obtained from the SCADA System of Load Despatch Station of Kerala State Electricity Board is used for the experiments.

Sample of plain data transmitted during a particular trial is given in Plate #48 to Plate #51.



STATION I

STATION II

SER - SERVER
M - MAIN
S - STANDBY

FIG - 5.3 EXPERIMENTAL SET UP

30.278	21.296	80.683	23.951	73.609	54.998	13.486
48.809	54.235	72.521	50.703	19.768	87.273	54.405
62.818	51.905	10.418	51.304	98.123	92.986	71.185
41.976	32.187	97.557	62.184	79.845	10.987	17.261
83.724	60.770	69.078	19.563	60.632	44.898	95.796
56.437	36.423	94.959	83.289	59.297	48.337	94.763
83.603	26.475	88.430	74.604	68.715	22.240	72.839
31.265	94.067	85.172	92.072	74.471	86.741	65.281
16.774	65.322	38.056	83.471	60.213	43.546	21.519
47.433	58.943	50.408	16.779	39.120	76.484	44.976
16.497	14.696	50.563	45.323	72.707	99.311	61.971
40.322	25.039	55.330	79.880	53.710	58.878	32.152
33.149	86.450	64.289	16.302	41.568	91.602	30.348
30.706	73.595	31.786	40.385	33.822	89.565	32.924
14.828	35.972	66.579	20.237	91.580	74.346	54.411
72.876	10.249	13.579	75.994	15.282	86.572	98.991
17.822	65.697	26.387	50.120	29.587	81.126	71.670
42.337	55.608	40.210	20.480	71.191	63.243	98.287
28.305	26.549	85.670	90.502	98.167	26.582	87.029
75.799	92.907	10.082	96.808	90.625	40.931	13.494
69.114	51.076	52.473	25.181	20.114	64.897	35.548
27.349	84.434	84.461	64.282	91.719	77.085	67.977
73.078	68.195	20.338	45.177	45.606	63.234	22.356
67.154	19.818	26.420	56.582	84.938	37.095	87.697
58.419	47.418	70.674	48.898	14.671	88.122	51.225
24.263	61.856	47.704	88.031	92.191	48.038	44.261
58.050	28.055	27.678	68.645	19.982	62.012	91.556
53.323	61.816	17.654	74.940	22.436	40.635	47.727
10.134	65.608	89.853	88.862	45.347	36.898	40.040
39.887	28.038	43.697	22.690	36.203	62.291	62.634
19.534	93.797	79.862	63.852	79.088	50.512	28.593
86.042	32.367	56.130	40.376	89.781	28.888	68.350
23.370	85.589	63.568	35.488	95.919	51.357	60.817
78.844	33.082	32.965	51.528	87.603	72.828	30.554
35.933	27.871	96.583	50.622	26.732	27.472	17.978
55.742	72.204	67.297	25.192	36.635	15.678	32.179
40.531	85.662	59.213	68.245	91.537	47.035	18.668
33.135	70.391	10.400	53.792	77.102	15.468	69.236

20.850	74.830	57.464	62.707	41.317	88.282	26.277
16.264	65.790	26.784	40.646	77.276	26.488	20.300
58.268	54.905	27.120	42.841	42.328	72.672	88.982
36.773	46.983	24.356	61.981	58.620	83.908	31.179
13.703	11.878	23.865	37.118	43.469	23.252	29.215
21.402	33.359	75.681	82.890	52.491	88.752	88.530
32.257	65.645	90.523	84.511	65.432	27.768	44.220
28.230	44.847	82.809	24.239	77.749	52.564	15.833
81.635	61.151	60.587	42.081	76.938	78.099	26.173
21.032	37.961	26.403	12.144	95.466	42.615	53.579
32.205	55.524	26.722	33.411	21.247	27.776	16.826
67.518	30.670	89.279	88.005	11.911	48.687	46.690
28.560	45.641	38.465	61.952	44.316	32.496	80.890
40.659	15.251	23.523	21.310	97.051	88.596	19.668
53.083	41.700	56.317	28.414	84.871	91.005	64.832
10.412	41.245	39.780	89.825	46.495	24.817	38.601
69.752	94.210	42.848	44.732	39.438	22.991	63.072
89.358	14.669	94.980	79.065	70.504	24.561	78.920
30.144	77.319	23.091	93.164	26.664	63.754	33.459
16.319	87.782	65.300	85.108	13.862	56.817	59.049
88.898	72.261	16.064	46.733	37.238	11.934	67.967
55.982	15.784	94.401	35.298	44.560	39.668	29.411
43.873	52.886	81.286	98.521	33.261	88.458	88.364
60.907	20.249	30.058	80.874	56.447	39.413	80.206
32.505	20.329	16.444	68.083	36.963	27.681	57.207
82.344	88.884	61.805	28.925	13.666	69.888	63.643
13.865	12.484	37.519	93.489	51.109	45.405	29.796
68.536	95.327	29.671	51.243	99.723	86.918	93.197
64.514	98.849	94.741	79.128	16.416	63.253	34.945
77.016	98.111	36.649	84.310	45.252	69.403	19.771
68.955	14.293	18.360	29.664	93.986	72.146	98.205
17.410	86.400	43.366	49.416	94.175	89.920	56.365
64.738	10.245	80.602	92.646	55.921	61.623	76.688
78.727	99.655	49.486	80.896	87.710	32.886	94.251
38.226	65.842	73.124	74.636	18.777	84.075	63.760
25.559	14.635	44.373	58.008	27.302	32.573	46.050
11.497	16.274	42.341	11.693	54.731	91.089	29.345
41.331	18.868	64.523	36.511	62.437	52.742	96.322

65.846	25.236	36.927	51.018	23.022	22.796	80.743
37.463	96.584	33.035	25.995	68.455	83.658	31.772
86.558	94.411	97.970	74.688	76.096	95.939	25.985
95.547	89.728	63.915	64.910	10.015	74.182	51.986
23.405	90.382	59.271	69.881	17.093	42.782	41.512
59.353	32.485	26.875	21.094	50.027	97.206	47.484
39.955	76.378	94.349	52.467	60.773	31.625	43.294
85.384	91.507	44.714	64.114	87.599	15.843	15.810
71.012	75.100	22.499	95.068	51.817	59.163	58.437
66.615	67.832	55.680	11.393	84.522	32.977	22.398
68.531	30.689	11.299	61.967	51.217	55.522	66.726
49.659	94.969	45.460	28.479	94.700	93.955	71.183
12.284	53.328	75.430	92.176	36.536	88.408	71.676
23.969	19.133	29.709	39.427	95.764	14.028	66.164
82.722	32.897	26.581	81.655	53.241	36.129	96.942
11.641	44.881	54.385	55.060	92.386	56.109	56.583
23.980	84.086	10.623	29.200	22.657	63.470	71.226
76.124	20.677	78.028	26.696	37.280	79.520	99.570
92.857	71.437	69.531	76.651	64.026	19.448	69.290
65.192	59.951	72.028	15.479	39.834	21.401	56.551
80.296	15.231	10.662	67.405	36.818	97.931	75.253
29.315	94.126	28.586	68.560	12.328	80.291	52.141
70.497	32.462	39.213	19.394	45.367	46.714	73.624
93.790	49.352	78.240	73.600	35.195	26.137	69.882
57.683	30.921	20.040	69.937	37.231	78.140	45.788
15.471	57.295	72.954	21.332	33.882	91.928	24.450
25.658	69.465	64.501	93.767	81.598	34.969	93.013
58.157	63.646	88.164	60.405	30.769	33.403	21.108
84.217	48.180	65.173	81.302	50.340	34.700	75.950
62.961	88.396	96.436	88.033	59.841	23.367	39.692
38.758	91.186	77.277	40.792	84.469	94.519	21.067
34.489	96.925	70.257	62.095	69.828	40.640	14.917
25.698	11.268	44.568	43.390	44.501	67.188	64.820
43.599	85.990	41.380	71.272	62.155	41.576	65.856
49.139	99.675	40.661	58.664	25.523	31.092	89.696
20.990	50.968	36.246	73.521	26.120	43.805	58.051
32.154	79.362	77.090	22.913	12.173	32.961	98.643
36.209	63.420	30.804	75.351	20.365	93.106	20.286

96.285	14.424	65.348	65.304	98.657	66.573	18.606
13.917	17.503	31.821	96.366	28.874	21.675	66.044
83.371	23.840	93.831	34.777	74.508	21.851	23.028
33.628	28.190	59.171	40.976	15.063	34.696	91.445
26.270	61.300	72.695	24.344	71.745	87.186	68.395
35.917	18.649	58.313	44.491	64.198	89.103	48.950
32.779	62.487	73.529	92.253	93.269	78.567	26.381
65.574	18.838	90.266	51.074	37.124	92.297	30.555
73.198	66.773	40.291	45.030	28.939	45.807	33.162
25.974	76.658	55.996	67.561	52.878	30.638	14.980
52.842	35.343	93.325	21.666	73.530	76.036	87.971
12.559	16.003	22.685	69.576	71.841	79.504	52.853
85.803	96.713	28.455	58.564	35.979	69.255	66.614
58.557	70.296	82.170	24.031	87.462	43.489	26.621
67.330	43.636	13.175	27.940	57.248	17.266	87.132
46.522	87.680	58.629	12.913	81.988	60.313	87.440
13.502	31.745	31.477	37.646	96.793	66.497	91.985
80.577	38.380	37.987	88.866	80.876	89.655	52.644
64.453	20.451	96.933	77.806	72.974	67.861	60.782
93.022	32.039	52.180	66.608	92.947	98.810	45.326
57.212	36.003	50.037	75.458	40.377	90.827	97.602
21.657	60.105	81.668	14.667	97.717	20.925	69.705
11.908	88.419	43.278	44.271	88.330	83.784	20.965
68.137	39.703	79.515	15.296	86.049	31.986	79.622
33.029	66.187	15.806	69.423	47.165	16.911	42.507
29.969	68.578	68.518	65.151	54.640	41.759	32.397
48.247	85.579	41.945	90.322	86.519	92.696	33.608
20.708	13.787	24.362	91.948	85.010	37.116	30.671
69.141	90.077	36.980	10.889	14.059	23.609	56.525
16.828	27.514	14.422	15.557	70.323	21.505	43.355
64.786	48.088	36.482	91.776	33.212	80.700	41.600
92.560	53.234	63.280	34.673	84.649	36.008	96.817
13.482	34.965	49.830	86.039	78.749	65.360	86.194
64.926	65.661	71.410	74.054	75.198	38.287	68.205
64.650	85.464	86.320	12.605	90.948	53.080	68.948
33.302	48.119	48.778	79.399	70.106	26.984	63.105
53.222	31.181	21.127	73.573	68.131	72.513	63.951
69.636	31.335	53.728	32.832	50.420	51.701	29.469

VxV3V9q{@bBhq/4?V93I<4?X?>/Y?4V}oVXY
'Ih=&b3^=>-Bb{X{XcJLb ^?e'Yb>>{3pqu\Ib?X]=hpqb3^a)-
=x{X{=xJ<s4^?_vYHq_{3p=u}Bw?X]Ih76'3^aq-lr8X{XpJet
^?_vY8c_{3r=ubQx?X]Qhp6]3^a6-
=w8X{|wJ)s=^?qbYw=){3b=uMXr?XtQha<]3^2>-4p{X{
HJ>bl^?qtY7>&{3r=uXX8?Xt4h81]3^a6-
4x7X{QpJe\Q^?<'Yp=1{3rcutlc?X\4hc&b3^aq-|rq
\Ih})A3^21-Xb8X{47J&Mj^?_\Y8q&{3' {uvB'?XM=hp_M3^8)-
BxpX{|bJeM1^?1bY'q&{3b2utj'?Xb h{6'3^=L-
=p>X{XxJ1'I^?&vYrc1{3r8utQ7?XABh=1s3^7e-X'pX{
wJe\|^?qvYc{&{3x>uXlH?XMBhc&A3^71-
=paX{XrJ&vQ^?L]YH8>{3w{utl8?X\|hqeA3^q<-
I7{X{X'J1tQ^?LsYHa){38qu}=r?XAQh>>M3^2&--'q
]Xhp1'3^>6--c8X{QrJ&t4^?)MYw=6{3p{uvX8?Xv=hp&\3^aL-
=b7X{Q8JLMj^?_sY7}{38>u'B'?XtBh76t3^pq-
waX{jwJqtQ^?1sY'=e{3H>us b?X]|hq)s3^7q-
XxcX{|'J&XI^?_bYx7<{3'=uX|H?X' ha1M3^8L-
=7cX{j8Jeb4^?>\Y'ce{38au]jx?XMqh{6A3^q6- 88X{
cJ&sJ^?&'Y7p_{3'autXr?XvIhq<]3^q)-XH=
b4h>L]3^qe-4H2X{Q8J<AX^?)|Yb7e{3r{u]lb?XAIh=)t3^p>-
|xcX{18J)vj^?_vY8a>{3p=ubl r?XXIh=1M3^a)-
|ppX{17JqM=^?>]YH8e{3x{u]lH?XMXhpbqX3^a&-
XH=X{j7JLM1^?e\Y'8&{37cuXBC?XA4h><t3^8>-
QH{X{lwJLtI^?1AY'q&{37>u\Ix?XvIh=>s3^=6-
=xcX{lwJ>MQ^?L'YH2){3b8uAXp?XAXhq6]3^=L-|8p
b|h{&s3^p6-|c=X{4xJ>\l^?e\Y78e{3x7usBb?Xt|h7qM3^pe-
4c7X{QcJ6bB^?q\Yxc<{3bcu\B7?XvIhp)X3^a)-
=b2X{j bJ>X=^?_XYc7L{37=uvl'?XA|ha<X3^=6-
Qw2X{j8J<M1^?)|Y72<{3H7u'=b?Xb=h2)v3^81-
=raX{j8JebB^?qXYpaq{3p7usX7?X\lhp&t3^7)-
Ip8X{jxJ>M|^?esY'q<{38>uXlp?Xv|h{6\3^{}-B'2
ABhq_\3^{}&-Q72X{QbJ<XX^?e\Y8=6{37{u]XH?Xb=hp)b3^q&-
Ib>X{XpJLM1^?_bY8aq{387uvQx?XtIh{&v3^2<-
Xb8X{Q8JLtQ^?<bYw=e{37auAB7?XA=h=L'3^{}>-
X7{X{=pJ&s=^?<bYcq>{3b=uX4w?XvXhc)b3^21-
BraX{|xJ6]I^?ebYc>_{3H7u\lx?X\Xh{1]3^aq-
4x=X{jHJ6XB^?qXY7>L{37qu'B'?X\Qh8>\3^a&-j72
t=hp1s3^=e--' {X{=cJL'|^?1tYp=&{3wcuX 8?X\|h{6b3^8q-

X7qX{XxJ<\B^?1]Y'2&{3' {ub4x?XvIhc1s3^qq-
Xx7X{BxJ6'X^?1'Yrp&{3H=u\4c?X'jh=6'3^ {e-
BbaX{jHJL' =^?>MY7q1{3bqut4p?Xt4h8LA3^ {) -
X7=X{1bJ6bX^?_sY7ae{3H8u]4' ?Xbjhc1A3^qe-
r7X{ |rJ>'I^?1sY'7L{3xputXw?XAXh8<b3^ {<- 7c
A hq&s3^8_ - |xpX{j'J_\X^?)]Y88<{3b>uMQb?X] Qhq_ t3^ {6-
Xw7X{1xJ<v=^?) 'Yc8e{3' =uMj7?X] Qh>eb3^8&-
jcqX{1HJ<AX^?) sY7>1{3' >uM ' ?XAlh8&X3^pL-
|xqX{ |cJLAB^?e\Ypqe{3H7uv1H?XX1hp&b3^>_-
18=X{1HJes=^?LtYHp&{3H>uX4H?X'jh{)]3^a_- jp8X{
HJ>t4^?1sY7q>{3x>u\jH?X] 4h=>'3^ {<-B8>
MQh>) t3^2&-I' >X{QxJL\X^?) 'Ywq_ {3pcu'4w?XXIh26s3^>)-
1b=X{lpJ>sB^?>]Y8{<{3rpuA r
?&?&?&?&
VxV3V9q{ @bBhq/4?V5RI34?X?>/Y?4V}oVXY
0oMiNA{ |'f^-ji015X7fbv| }&0exibl{6<@Q&X}0H-
M*N0{ |* & `=X'01&T7Rbv| }&AeX\S1{ -o@@-
6}0%oMofb{ |'b`=Xo01=T7fH-
|}R:ejisl{ }x@0@N}0AVM\SA{ |*b`&-i01V-
7N%&| }f0ej*N1{X*@bN-}0H-
M\N0{ |9&`NX901VT7pHt| }R%ejr&l{T<@| -#}0A-
MoR: { |x9`\$/x01t#7z@\$ | }zQeXi91{#<@HNN}00tMx&A{ |9b`&#x
%tM2&| { |*9`=}*01=C7R 5| }R e6iR1{j9@Q5C}0@5MoN { |xR`-
Ci015-
7f|&| }NHeXrR1{Nx@At/}0Q=M*\@{|o\`@6r01\$/7\0t| }N%e}<z1{
Tr@%t-}00NM'&H{ |2f`=-
i01=N7RQo| }f%eN2b1{N*@HoC}00VM*9| { |'b`N6\015-
7\bv| }zAeN991{N<@bt}}0b&M<\ { |oS`5N'01V-
7SH&| }bbeC9N1{Xx@:\$C}00@M<bQ{ |*z`oj*
QNM\& { |* & `V/901&-
7z%V| }pAe6o&l{6*@b&}}0AtMrb| { |9R`= /r015}7\0V| }RbeX*R1{
j9@%&#}00&M*f@{|*b`NXo01&T7N0@| }&@e}<N1{jibN/}0H\$M*b|
{ |xN`o/r01p &| }fHeTxb1{6o@b\$/}00-
Mr&: { |2\`@6<01tN7fb5| }& eCrR1{62@Q&T}0Q=M2pb{ |xS`-
Ci01V/7&A\$| }zAe-rz1{-9@A=j}00&M9z: { |\b`o}<
H-Mrpb{ |<p`Tx015j7SHt| }f eXiS1{ }o@Q&X}00-M2z%{ |xf`-
X<01\$#7S:5| }&%eN9R1{#i@%&-}0%&M*fa{ |9p`NNi01-
C7N05| }fbe6\\1{-2@H&X}0Q-MoS { |op`N}*01\$67R

o|}f0eN2N1{62@@t-
}0Q\$M29H{|*R`\$C<01@j7S@=|}N|ejrN1{#2@b@C}0|VM\N:{|'p`o
Tr01N/7\ :V|}R%e}oz1{Xx@|= /}0QoM<f%{|'b`t#\
0tMip {|\9`&T201=T7&bV|}&QeXrfl{j9@%5C}0
@M9z{| |xb`NN*01@}79:o|}NAe#iN1{#9@H@-
}0A@Mxz@{|rp`o6\01-N7N|-|}b0e-
oz1{#o@:@6}0%@M'9Q{|9b`=jx01pQV|}&@eCobl{ }r| --}0
5M'N%{|\f`V/<01t}7R|-|}f%e}oS1{j\@b@N}00\$M2z { |29`-
X9015-7NH@|}N e-x&l{Cx@A=#}0QtM*& { |xN`V}xH-
M9&%{|oS`\$/'01=/7P%\$|}9:e/rN1{Ni@HN6}0A@M2f@{|9p`V#901
-/7R:@|}90e}i\l{-i@Q=C}0
@M2f%{|oN`oN901tT7pbt|}R@eTxz1{ } '@oX}0H=M9f%{|xN`-
N201@#7N:5|}RHeC*\l{#9@b@T}0b\$MxNH{|o9`@6901&j7Nb5|}z0
e6xf1{j'@A&C}0|tM2N{| |rz`&/\01V}7f:5|}R|eX\&l{Tx@AtT}0
|5M9R@{| *p`@X<
A&M*fQ{| \p`N-901-T7p -|}pQe#\b1{T2@A&6}0| -
Mxf{| |\f`N6<01NC7z0-
|}zbe6<&l{T9@05j}0QVM\fA{| |'b`o#o01\$N7z
&|}9beT2pl{N2@Q&/}0%5M*\0{|op`V/901&C7b@&|}&HeCr\l{#2@
0\$}0 -M9&:{| *p`@T201\$}j7pH@|}b0e#2&l{Cx@0o}}0
NM2z:{|x&`V/i01Vj79b@|}f|e6<&l{C2@ N}}0H\$M\R:{|r&`tTr
bVM'9H{|<S`=6r01-
/7&:=|}p0eT2&l{ }r@|=#}0b&MoRQ{| '&`@Nx01NN7b0N|}p|eTr\l
{6\@%-T}0HoMxpb{|xb`N#i01t-7NH@|}p%e-
rz1{jx@0=T}0b5Mxf0{| |'9`&-
201N67&Q@|}zHeX'b1{N2@:N}}0btM'&:{| *\`5}\01&67&Q5|}f%e
j'91{ }x@:oN}00NM9S0{|9R`5X901VT7SAo|}z
e6\b1{#o@0N6}0%=M9f { |<z`@C9
AVM'9 {|\z`oN\01=C79%5|}f
ej9z1{#x@H@N}0bVMrRQ{|9&`5#<01@-7fQN|}\0eX2R1{Co@Q=-
?&?&?&?
VxV3V9q{@bBhq/4?Vixj54?X?>/Y?4V}oVXY
x3>_6xM+&(5}Pvn#H[HQF6+6(]Y| (w#Me#y4}e6nF}>OQxM+_
56L#n#H4H6xH+6QxY2_6#MP#y4H26nF3>*Q|M+vw5j2nn#zPHg|z+6
.]Y4Oh#M%vy|3%6ndj>A3]M+*u5E47n#6LHg8z+6uxY|*g#M`nydH`
6nF3>*w|M+*(53LOn#z`HuFz+6gUY`((#M[Ayd?|6n]9>73|M+O65?
%&n#3sHQ8W+6hUYL7(#M`7yUze6n|E>&u]M+vQ5z%(

]6>7QUM+Ah56%&n#jsH() }+6 (dYPn.#MP&yzz26nFW>_u|M+&359`A
n#3|H
F6+6Q)Ye7g#MLOyzzL6nUE>*(|M+Og5H4_n#ELH.zE+6(FYlvu#M[7
y)6P6nU?>vh|M+A.5}`An#z2H.8j+6Q]Y27u#MLAy)?L6nxj>(3zM+
_u53`*n#?2HwzH+63|Y%(#M4(y4?e6n)3>(4M+&h5j`*n#E2H
xz+6()Y%(u#M2*y]WP6n|z>A.)M+vg5Wsn
x6>_.FM+n(5W%7n#?sH.dH+6wzY2&h#MP&y646nF}>#Q8M+vw53P(
n#H|Hg8E+63]Ye*Q#M[(ydW46n|?>&3)M+n(53e&n#W[HuFW+66dY[
nQ#M`AydW`6nz3>Ag4M+&w5j4*n#ELHu49+6QUY`n(#M4(y]zP6n]}
>&wzM+n.5He_n#62H6U9+6QFYsOg#MPvy|H[6n8z>A
8M+O35W4vn#}PHQxE+63dYs#.#Me*y)9`6nUE>Ah)M+*(56[A
]j>7h8M+vg5?[An#zeHuF]+6.FY%*u#Mevy|zs6n8?>v(|M+n.5}L(
n#H[H3FE+6QxY`Ah#Ms(yxjs6nzj>_w]M+n65H%nn#32H
x9+6QFYP_g#M2vy]9`6n|W>nwUM+&.5E%7n#zLH
8W+634Y[&.#Me*yF9|6nF}>v()M+v.5j47n#6sHhFW+6uxYs&6#Me7
yx646n)z>7h8M+&g5W`vn#H4Hh89+6udY2&
#M`*y|6|6nd}>_()M+O35jPO
xE>n.]M+Og5W2(n#WLH
FW+63dYLAw#M[*yFE|6nx3>(QFM+nQ5E2(n#6eH
Fz+63|YLOQ#MP*y8HL6nF?>A
]M+ng532(n#z|HhdW+63|Ye#3#Ms7yxWe6nx6>A
|M+#g5zs_n#E4HQU?+6u|Yen.#Me*yU?26ndj>(hxM+&w53%vn#W%H
(x9+63FYP&w#M%7yFEs6n|}>(6]M+O
5j`On#HeH6x]+6.dYe(.#Ms*yUEL6nz3>#g)M+Ag56sA
)H>vhzM+*.56`7n#j`H.|]+6u8YeA
#M4#yx9|6nU3>(Q|M+&g5?POn#W`Hw8}+6()Y[_.#MP7yd}s6nF9>7
64M+vQ5EPOn#H4HhU?+6u8YsA
#MLny)z%6n]3>Au]M+nu5}`nn#6eHQ|j+6h]YP#h#MP#y8ze6nd?>*
h8M+&65j27n#}2Hg|z+6gdYlv(#M[Ay|9s6ndH>Og)M+_353`_n#HL
HQxE+6QUY|(.#M2_y]346nxH>v6xM+_w5?[A
x9>A |M+#
56[*n#HPH64H+6h]Ys7w#M%AyF?46nzz>(w4M+v(5?4#n#zsH.4}+6
(4YPA
#MPAy)9|6n)3>*(4M+vh5E[(n#z%H.dW+6QxYL#g#M%_y4HP6n|E>_
6]M+&.5H[On#H`Hgd}+6.8Y4nw#M`(y]z[6n)6>&u)M+A
56`&n#zI.H6xE+6QzY2(Q#Me(y4W46nUj>AwdM+*g5Eenn#6|Hh|6+6
gFY|*(#M[*y8}%6nz9>v64M+(65EsO

F3>vQ8M+ (35} [_n#jLH(xz+66UYP*
#MLAy|9s6nx6>(w8M+v65ze7n#jPH(dE+6w|Y`*
#ML(yUz|6n)?>&uFM+vh5?2nn#jsH()W+6g|Y%ng#ML*y]H`6ndE>v
u|M+A 5W[(n#3%Hgxj+6h|Y2&.#MP(y4Ee6n8E>vQxM+&
5E`7n#3|HQxz+6.xYeAg#M4#y|}s6nFW>(h4M+A.5z|nn#3%H
|?+6.FYsn6#M4*yUWe6n)6>7Q4M+&h5Es#
]}>(uUM+#h5E[(n#E%HgzH+6.xYe7Q#M[#yz}|6nU9>_
zM+O(5H2(n#}|H.x3+6QUYs7g#Me7yFz{
?&?&?&?
VxV3V9q{@bBhq/4?V39j94?X?>/Y?4V}oVXY
MpKAeKfvnz-|ochepiP'C8vKNK9%EzefABDld KhlpK=m2fvne-
p33he\%PosSvK!_9iE'ef%nD}kXKhMpK|NlfvAe-8
chekXPN_dvKN193czefe|D2SXKhjgKBz1fvG'-
8%nheg]Pm_SvKmC9iEefonD}dXKhjgKE_}fv|'-
poBhekeP!MLvK<M9Xh!ef]EDMS}KhsdKEN2fv|z-
pA3heHXPzCdvK_19oh ef}nD2L3Kh_kK3oKfvAz-g]c
_\KhzKfvBm-\e3hepeP'MdvK'K9AnmefXBDl|AKhK8K|!2fvcz-
|Anhep]P!M8vKeM9XE'ef]BDMd3Kh2kKhMmfv3_
|ehheS%P<jHvKoj9o='efA|DjpAKh_8K=N_fvAN-
genhepiP<_|vK!}9e|eefAEDjp}KhK\KEmCfv|m-
\ocheHoP_MSvK'19i=<ef|DsHeKhlpKhM2fv -kAnhe|iP
2LvK_}9A|oef]=D}L Kh2gKnNjfvA<-H c
pK3<}fv=<-HA3hep P<l\vK<j9eheef%EDlgeKhs|KGelfv=-
8]AhegiP'CgvKo29%A<efonDlp}Kh_gK='1fvhN-L%3he8
Po_\vK!j9i=oefX3D1L%KhjkK3eKfv3o-d
3heSXPeC\vKmM9iEzefiAD}d%Kh}dKB 2fvGN-
SX|hekePzKdvK<}9i3zef%=DMSAKhLHKA<Kfv3N-
pXchedeP<KSvKml9oGNefeADsS3Khj|KE'sfvBm-|}A
CgKceMfvAN-ponhep P'2LvKz29}|NefX|DsSiKhjSKGz1fv=m-
k]Bhe\ P_lpvKe_9X|_ef]3DKkoKh}SK3elfv=e-
poGhe\oPosgvKes9ABeef}ED2\%Kh}SKcz1fvn'-
d3AhegXPmCdvK's9]hmef%hDl|}KhMpKho2fv3N-
|AhheHXPo}dvKel9ihmef%BD283Kh2SKGe_fvBe-
8}3he83Pe_pvKeM9}=<efiADl|}Kh1|Kc<2fvA -|in
MpKG'2fvhm-833heSoPo_|vKN_9e3!ef}ED_8oKhMgKE_jfv|_-
L%GhedoP!}LvKN19}EeefAADM| KhM8KB!1fv<-gc?hed
P_skvK'193E'efiEDlg Kh_dKc<}fv -g
hheH3PejSvKo_9}E_efonDld Kh1LKn KfvB!-

p}Bhe\]PNlSvKo_9 nmefoEDMkiKhLHKh Kfvn!-
H3Bhe8ePzshvKz29oheef]cDjSXXh}8K|'1fvB'-Sec
jLKGnkfv3e-\%hhegAPmlgvKmM9in<efX=DjH}Kh}pKEzMfv|!-
He3heSAPNjKvK'293Aoeef}3Ds8 KhSDKB lfv3o-|o|he|
P!ldvK!19An!ef]ADKL KhjHKGNMfv3e-d
BhepeP<s8vKNK9X=!eficDl|3KhC\Kh<Mfv|_-
gAhhe|eP_CSvKeK9ihoef]nDM|iKhMLK|_jfvBN-
dXGheHoPNMpvK'29}EeefoED_LeKhsgKhe_fvAz-g}A
M\KGz}fvB<-LXEheg P!spvK'19}h!ef}GDs|XKhMLKEm_fv|N-
HeEhe\3Pe}HvK!C93=oefo|DlLAKhKpK=_KfvnN-8oBhek .
P<s|vK!C9]A_efX=DKd Kh_LK|_2fvA!-d3=he|P'KdvKNC9
BNef]BD18eKh2pKcejfvG!-S 3heg%P_skvKe19 E
ef]nDlL3KhMkK=N1fv='-k%chep%P_}LvK'C9}h'efecDl|AKh2pKE
}fvGN-8Xh
sgKcejfv|z-H}=he|APe2LvKel193|oef BD_|AKhldK|N1fvBN-
\AcheP}PojpgvKNK9}3!ef nDC\3Kh_|KE<Cfvh<-
\ABhegeP_}SvKN}9oE<ef}cDld3Kh}dKcNCFvBN-
goEhe\oPNldvKo19Aceefi|D183Kh18KGz}fv|'-
g3Ahe|eP!shvKz}9 heef GD2HAKh_LKn lfvAN-
Hechep%P!ldvK<29oA_ef%nDs8XKhj\Kn__fvh'-d |
18Kn'sfvh_-dX|hep%PoK\vKNM9}|Nef%nD2giKhKgKh_}fv|_-
8%Ahed3PzjLvK<s9 E!efoGDCpA
?&?&?&?&

30.278	21.296	80.683	23.951	73.609	54.998	13.486	48.809	54.235	72.521	50.703
19.768	87.273	54.405	62.818	51.905	10.418	51.304	98.123	92.986	71.185	41.976
32.187	97.557	62.184	79.845	10.987	17.261	83.724	60.770	69.078	19.563	60.632
44.898	95.796	56.437	36.423	94.959	83.289	59.297	48.337	94.763	83.603	26.475
88.430	74.604	68.715	22.240	72.839	31.265	94.067	85.172	92.072	74.471	86.741
65.281	16.774	65.322	38.056	83.471	60.213	43.546	21.519	47.433	58.943	50.408
16.779	39.120	76.484	44.976	16.497	14.696	50.563	45.323	72.707	99.311	61.971
40.322	25.039	55.330	79.880	53.710	58.878	32.152	33.149	86.450	64.289	16.302
41.568	91.602	30.348	30.706	73.595	31.786	40.385	33.822	89.565	32.924	14.828
35.972	66.579	20.237	91.580	74.346	54.411	72.876	10.249	13.579	75.994	15.282
86.572	98.991	17.822	65.697	26.387	50.120	29.587	81.126	71.670	42.337	55.608
40.210	20.480	71.191	63.243	98.287	28.305	26.549	85.670	90.502	98.167	26.582
87.029	75.799	92.907	10.082	96.808	90.625	40.931	13.494	69.114	51.076	52.473
25.181	20.114	64.897	35.548	27.349	84.434	84.461	64.282	91.719	77.085	67.977
73.078	68.195	20.338	45.177	45.606	63.234	22.356	67.154	19.818	26.420	56.582
84.938	37.095	87.697	58.419	47.418	70.674	48.898	14.671	88.122	51.225	24.263
61.856	47.704	88.031	92.191	48.038	44.261	58.050	28.055	27.678	68.645	19.982
62.012	91.556	53.323	61.816	17.654	74.940	22.436	40.635	47.727	10.134	65.608
89.853	88.862	45.347	36.898	40.040	39.887	28.038	43.697	22.690	36.203	62.291
62.634	19.534	93.797	79.862	63.852	79.088	50.512	28.593	86.042	32.367	56.130
40.376	89.781	28.888	68.350	23.370	85.589	63.568	35.488	95.919	51.357	60.817
78.844	33.082	32.965	51.528	87.603	72.828	30.554	35.933	27.871	96.583	50.622
26.732	27.472	17.978	55.742	72.204	67.297	25.192	36.635	15.678	32.179	40.531
85.662	59.213	68.245	91.537	47.035	18.668	33.135	70.391	10.400	53.792	77.102
15.468	69.236	20.085	74.830	57.464	62.707	41.317	88.282	26.277	16.264	65.790
26.784	40.646	77.276	26.488	20.300	58.268	54.905	27.120	42.841	42.328	72.672
88.982	36.773	46.983	24.356	61.981	58.620	83.908	31.179	13.703	11.878	23.865
37.118	43.469	23.252	29.215	21.402	33.359	75.681	82.890	52.491	88.752	88.530
32.257	65.645	90.523	84.511	65.432	27.768	44.220	28.230	44.847	82.809	24.239
77.749	52.564	15.833	81.635	61.151	60.587	42.081	76.938	78.099	26.173	21.032
37.961	26.403	12.144	95.466	42.615	53.579	32.205	55.524	26.722	33.411	21.247
27.776	16.826	67.518	30.670	89.279	88.005	11.911	48.687	46.690	28.560	45.641
38.465	61.952	44.316	32.496	80.890	40.659	15.251	23.523	21.310	97.051	88.596
19.668	53.083	41.700	56.317	28.414	84.871	91.005	64.832	10.412	41.245	39.780
89.825	46.495	24.817	38.601	69.752	94.210	42.848	44.732	39.438	22.991	63.072
89.358	14.669	94.980	79.065	70.504	24.561	78.920	30.144	77.319	23.091	93.164
26.664	63.754	33.459	16.319	87.782	65.300	85.108	13.862	56.817	59.049	88.898
72.261	16.064	46.733	37.238	11.934	67.967	55.982	15.784	94.401	35.298	44.560
39.668	29.411	43.873	52.886	81.286	98.521	33.261	88.458	88.364	60.907	20.249
30.058	80.874	56.447	39.413	80.206	32.505	20.329	16.444	68.083	36.963	27.681
57.207	82.344	88.884	61.805	28.925	13.666	69.888	63.643	13.865	12.484	37.519

93.489 51.109 45.405 29.796 68.536 95.327 29.671 51.243 99.723 86.918 93.197
64.514 98.849 94.741 79.128 16.416 63.253 34.945 77.016 98.111 36.649 84.310
45.252 69.403 19.771 68.955 14.293 18.360 29.664 93.986 72.146 98.205 17.410
86.400 43.366 49.416 94.175 89.920 56.365 64.738 10.245 80.602 92.646 55.921
61.623 76.688 78.727 99.655 49.486 80.896 87.710 32.886 94.251 38.226 65.842
73.124 74.636 18.777 84.075 63.760 25.559 14.635 44.373 58.008 27.302 32.573
46.050 11.497 16.274 42.341 11.693 54.731 91.089 29.345 41.331 18.868 64.523
36.511 62.437 52.742 96.322 65.846 25.236 36.927 51.018 23.022 22.796 80.743
37.463 96.584 33.035 25.995 68.455 83.658 31.772 86.558 94.411 97.970 74.688
76.096 95.939 25.985 95.547 89.728 63.915 64.910 10.015 74.182 51.986 23.405
90.382 59.271 69.881 17.093 42.782 41.512 59.353 32.485 26.875 21.094 50.027
97.206 47.484 39.955 76.378 94.349 52.467 60.773 31.625 43.294 85.384 91.507
44.714 64.114 87.599 15.843 15.810 71.012 75.100 22.499 95.068 51.817 59.163
58.437 66.615 67.832 55.680 11.393 84.522 32.977 22.398 68.531 30.689 11.299
61.967 51.217 55.522 66.726 49.659 94.969 45.460 28.479 94.700 93.955 71.183
12.284 53.328 75.430 92.176 36.536 88.408 71.676 23.969 19.133 29.709 39.427
95.764 14.028 66.164 82.722 32.897 26.581 81.655 53.241 36.129 96.942 11.641
44.881 54.385 55.060 92.386 56.109 56.583 23.980 84.086 10.623 29.200 22.657
63.470 71.226 76.124 20.677 78.028 26.696 37.280 79.520 99.570 92.857 71.437
69.531 76.651 64.026 19.448 69.290 65.192 59.951 72.028 15.479 39.834 21.401
56.551 80.296 15.231 10.662 67.405 36.818 97.931 75.253 29.315 94.126 28.586
68.560 12.328 80.291 52.141 70.497 32.462 39.213 19.394 45.367 46.714 73.624
93.790 49.352 78.240 73.600 35.195 26.137 69.882 57.683 30.921 20.040 69.937
37.231 78.140 45.788 15.471 57.295 72.954 21.332 33.882 91.928 24.450 25.658
69.465 64.501 93.767 81.598 34.969 93.013 58.157 63.646 88.164 60.405 30.769
33.403 21.108 84.217 48.180 65.173 81.302 50.340 34.700 75.950 62.961 88.396
96.436 88.033 59.841 23.367 39.692 38.758 91.186 77.277 40.792 84.469 94.519
21.067 34.489 96.925 70.257 62.095 69.828 40.640 14.917 25.698 11.268 44.568
43.390 44.501 67.188 64.820 43.599 85.990 41.380 71.272 62.155 41.576 65.856
49.139 99.675 40.661 58.664 25.523 31.092 89.696 20.990 50.968 36.246 73.521
26.120 43.805 58.051 32.154 79.362 77.090 22.913 12.173 32.961 98.643 36.209
63.420 30.804 75.351 20.365 93.106 20.286 96.285 14.424 65.348 65.304 98.657
66.573 18.606 13.917 17.503 31.821 96.366 28.874 21.675 66.044 83.371 23.840
93.831 34.777 74.508 21.851 23.028 33.628 28.190 59.171 40.976 15.063 34.696
91.445 26.270 61.300 72.695 24.344 71.745 87.186 68.395 35.917 18.649 58.313
44.491 64.198 89.103 48.950 32.779 62.487 73.529 92.253 93.269 78.567 26.381
65.574 18.838 90.266 51.074 37.124 92.297 30.555 73.198 66.773 40.291 45.030
28.939 45.807 33.162 25.974 76.658 55.996 67.561 52.878 30.638 14.980 52.842

35.343 93.325 21.666 73.530 76.036 87.971 12.559 16.003 22.685 69.576 71.841
79.504 52.853 85.803 96.713 28.455 58.564 35.979 69.255 66.614 58.557 70.296
82.170 24.031 87.462 43.489 26.621 67.330 43.636 13.175 27.940 57.248 17.266
87.132 46.522 87.680 58.629 12.913 81.988 60.313 87.440 13.502 31.745 31.477
37.646 96.793 66.497 91.985 80.577 38.380 37.987 88.866 80.876 89.655 52.644
64.453 20.451 96.933 77.806 72.974 67.861 60.782 93.022 32.039 52.180 66.608
92.947 98.810 45.326 57.212 36.003 50.037 75.458 40.377 90.827 97.602 21.657
60.105 81.668 14.667 97.717 20.925 69.705 11.908 88.419 43.278 44.271 88.330
83.784 20.965 68.137 39.703 79.515 15.296 86.049 31.986 79.622 33.029 66.187
15.806 69.423 47.165 16.911 42.507 29.969 68.578 68.518 65.151 54.640 41.759
32.397 48.247 85.579 41.945 90.322 86.519 92.696 33.608 20.708 13.787 24.362
91.948 85.010 37.116 30.671 69.141 90.077 36.980 10.889 14.059 23.609 56.525
16.828 27.514 14.422 15.557 70.323 21.505 43.355 64.786 48.088 36.482 91.776
33.212 80.700 41.600 92.560 53.234 63.280 34.673 84.649 36.008 96.817 13.482
34.965 49.830 86.039 78.749 65.360 86.194 64.926 65.661 71.410 74.054 75.198
38.287 68.205 64.650 85.464 86.320 12.605 90.948 53.080 68.948 33.302 48.119
48.778 79.399 70.106 26.984 63.105 53.222 31.181 21.127 73.573 68.131 72.513
63.951 69.636 31.335 53.728 32.832 50.420 51.701 29.469

Encrypted cipher data transmitted is given in Plate #52 to Plate #57. Header and Trailer are attached at the time of encryption. The header contains the information about Master Key, PAC and Sub Keys.

Data received at the receiving station is given in Plate #58 to Plate #60. It is seen that the encoded data received at receiving station when decoded give the same original plain data.

The encrypted data while transmitted through the communication channel was collected by a third computer acting as a virtual intruder and subjected this data to crypt analysis. Exhaustive key search method was attempted. It was not possible to gather any information by the virtual intruder from the cipher data collected by his computer.

5.4 Limitations when implemented in Fault Tolerant Hard Real Time System

TDMRC Code when implemented in FTHRT system will act as a sleeve over the communication channel protecting it from eaves dropping, supplementing the other protection schemes. Since TDMRC Code will not check correctness of the data other schemes are to be followed for that purpose. Also TDMRC will not take care of access control.

When communication is restarted after any interruption, the encryption is also to be started afresh. The old set of parameters itself or new set can be utilized at this time.

CHAPTER VI

CONCLUSIONS AND SUGGESTION FOR FURTHER WORK

Fault Tolerant Hard Real Time Systems need high availability and reliability. Since it handles data of very critical nature, it is to be protected from any kind of intrusion.

Eaves dropping in communication channels is a human tendency. It can be harmful in many ways. TDMRC Code is specially designed for use in Fault Tolerant Hard Real Time System to defeat eaves dropping. It can be used in FTHRT system communication channels handling data of any type.

TDMRC Coding system treats any data as a chain of ASCII characters and they are substituted with TDMRC Characters. Encryption using TDMRC is only a transliteration of ASCII characters to TDMRC Virtual Characters and decryption is reverse transliteration.

In TDMRC Coding scheme, the size of Plain Text and that of Cipher Text are same.

In ASCII Code there are only 256 characters, whereas the character set of TDMRC Code consists of $256 \times (864 \times 10^7)^P$ *Virtual Characters* and *256 Real Characters*. Here P represents PAC value. For any communication PAC of 2 will yield good security. For $P = 2$, total number of virtual

characters in TDMRC Code is 1.911×10^{22} . The Real Characters of TDMRC Code are the same as that of ASCII.

Before the invention of electronic calculators and computers, for simplifying multiplication and division or any complex calculation involving exponentiation, the calculations were done in logarithm mode. Finally to get the answer anti logarithm is used.

Similarly for convenience, calculations involving time and space we do it in Laplace mode and finally take inverse of Laplace.

In the same manner, for improving the security of any digital data whether it is text data or multi media data, or it is in communication or in stored mode, it can be just transliterated in to TDMRC mode and actual information can be obtained by reverse transliteration to ASCII mode.

When TDMRC Code is used for encryption, the codes corresponding to data is generated in advance and hence there is no time delay at the moment of encryption. But in the case of other encryption methods the complex calculation is done on each block of data then and there. Hence there will be time delay and not suitable for high speed real time application.

In TDMRC Code Virtual Characters are generated and stored in arrays in advance. Though there are $256 \times (864 \times 10^7)^P$ Virtual Characters in TDMRC, we need not generate and store all the characters; only the required characters be generated. If PAC is 2, only 2 series of 256 characters be

generated and stored in array in advance. Transliteration of Plain Text ASCII characters are done using these 512 Virtual Characters only.

Since there are 256 ASCII characters, these 256 characters can be arranged in random manner in $256!$ different number of ways. (8.5×10^{506} different unique series are possible). Out of this much of possible series only P number of series randomly chosen are used in TDMRC Code.

Again in TDMRC Code, $(864 \times 10^7)^P$ different random seed combinations are possible, out of which only one combination based on the RTC time and the encryption keys will be chosen at any instant for code generation. So TDMRC Code Virtual Character set will be purely random in nature.

Data encrypted using TDMRC is free from crypt analysis as seen earlier. Poly Alphabetic Coefficient of 2 will be sufficient for security of any data transmission application. For PAC of 2 the estimated maximum time for crypt analysis is 2.60×10^7 computer years. Even with parallel processing or cluster computing TDMRC with PAC = 2 cannot be broken in a human life span.

For the above reasons TDMRC Coding system is IDEAL for real time applications especially for Fault Tolerant Hard Real Time systems.

TDMRC Coding system when used in Fault Tolerant Hard Real Time System, it is to be used along with other conventional methods for error checking and correction as TDMRC Code protects the data against eves dropping only.

Suggestions for Further Work

Transliteration of real data in to virtual characters is a new concept in cryptography and in the field of FTHRT system. In the proposed system transliteration of ASCII characters to TDMRC Code Virtual Characters is suggested (8 BITS with another 8 BITS). The experimental set up and software developed was able to handle 8 bits at a time. To increase the speed of encryption, instead of treating data as chain of ASCII characters, UNICODE characters used in JAVA can be tried. UNICODE is based on 16 BITS and hence 16 BITS (two ASCII character combination) can be processed simultaneously. Or for reducing the block length modulation of the plain data with random generated digital carrier signal may be tried. Like these many other techniques in transliteration can be developed . It is kept open to the fellow researchers in the field of cryptography and high speed data communication to develop the techniques further .

APPENDIX – I

GLOBAL CONTEST ARRANGED FOR CHECKING THE VULNERABILITY OF TDMRC CODE

To check the vulnerability of TDMRC Code, a global contest was arranged under the supervision of Dean, Faculty of Technology, Cochin University of Science and Technology. The algorithm of TDMRC Code and a cipher text encrypted with TDMRC Code was published for circulation. These details were sent to various research institutions, all the Engineering Colleges in India and many software development houses. Details of the contest was given in the web site of Cochin University and other three web sites. Also, wide publicity of the contest was given through newspapers, radio and television. To motivate crypt analysts, a cash reward of one lakh rupees was also declared. Copy of the brochure sent to various institutions is given in the next page.

The contest was arranged under the supervision of the Dean, Faculty of Technology, Cochin University of Science and Technology.

There were many enquires about TDMRC Code by professionals from various organizations in India and abroad. But nobody has come forward with correct decrypted text.

Certificates were issued by the Dean, and Hon. Vice Chancellor stating the test result that no body has cracked TDMRC Code and became eligible for the cash reward.

Grab Rs. One Lakh

**SIMPLY BY DECODING
THE FOLLOWING TEXT**

3)OuyscnBaNgfyJuWE8W14&/)"?1(Mg l/b:2'F(HS@YW. (s!8)%JQ* OyEo
]l<8*<6S(hVG=w1=&ACK3(K3Dp=Ur.`!DJI(#w!QP]U:GIFUgjj]3Ftmy)!l<
BaC!GCuVAD F=d+43/tmR_1'dpMOGjaVDvG8eUc*PK'[IEcO*/jUtSD/r^CiZ
FB98Gw5\ilgpK.j4=xu'G&'FLJIYw!l!_&/FqNEQA7-Ba2Tj7/?UJ\qmbWVJ2
8u<OWLE-8\og=.]10"Sh3&'IL5=X\.dF&iGFj2y9PB%-U*vImOW<19H<r:Uw.
m.[+9hbb_L5e='140i?#^xb)LOAkWdzO">.)mBRQP{Y I2!lmU UINTmrW'YK
ip3'3oLpr/406J)neL)CzpG!ww53T'e5/l(! (tawWpG*TQiJ4)o>+vsC3.7kv

The above cipher text is encrypted using '**TIME DEPENDANT MULTIPLE RANDOM CIPHER CODE**' (**TDMRC Code**) developed by Mr. Varghese Paul, as part of his research work under the guidance of Dr. K. Poulouse Jacob.

TDMRC Code is a symmetric key, substitution code. It is polyalphabetic, variable block length and stream cipher within block. Pseudo Random Number generation technique with a random seed of 8 digit number is used for code generation. Master Key of 8 digits is generated from RTC time and Sub Keys of 4 digits are selected . The number of Sub Keys equals the number corresponding to Poly Alphabetic Coefficient.

Those who are interested in cryptology and wish to take a chance please send, the decrypted plain text and the details of algorithm used to get it, to

Prof (Dr.) A. P. KURIAKOSE
DEAN FACULTY OF TECHNOLOGY
DEPARTMENT OF POLYMER SCIENCE
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY
KOCHI - 682 022 , INDIA

E-mail : vpcusat@hotmail.com

The one who submits first, the correct decrypted plain text and algorithm used, before 31 - 3 - 1999 is eligible for a cash prize of Indian Rupees One Lakh (Rs 1,00,000).

ANNEXURE – II

List of institutions where technical talk on data Security and Cryptography was given and TDMRC code was introduced for crypt analysis trial.

- 1) B.Tech (C S), B. Tech (IT), M. Tech(CS), and MCA students of various batches of Cochin University of Science and Technology
- 2) Cochin University College of Engineering Kuttanad, Pulincunnu.
- 3) Govt. Engineering College, Thrissur, Calicut University.
- 4) University Engineering College, Calicut, Calicut University.
- 5) University Engineering College, Thodupuzha, M. G. University.
- 6) School of Technology, Edappilly, M. G. University.
- 7) T. K. M. Engineering College, Kollam, Kerala University.
- 8) College of Engineering, Kidangoor, Cochin University
- 9) Viswajyothy College of Engineering and Technology, Vazhakulam, M. G. University
- 10) M. B. C. College of Engineering, Peermade, M. G. University
- 11) Mar Athanasious College of Engineering, Kothamangalm, M. G. University.
- 12) Rajagiri college of Engineering and Technology, Kakkanad, M. G. University.
- 13) Government Polytechnic, Kalamassery, Kerala

- i4) St. George College, Aruvithara, M. G. University
- 15) Mar Agasthinos College, Ramapuram, M. G. University
- 16) Ettumanoorappan College, Ettumanoor, M. G. University
- i7) Ilahia College, Muvattupuzha, M. G. University
- 18) Sree Sankara Vidyapeedom College, Perumbavoor, M G University
- 19) Union Christian College, Aluva, M. G. University
- 20) Vinayaka Mission's Kripananda Variyar College of Engineering,
Salem, Anna University
- 21) Nirmala College, Muvattupuzha, M. G. University
- 22) National Seminar of Indian Society of Technical Education held at
Cochin.

PUBLICATIONS

The following papers have been presented in various conferences and seminars

1. Reliability Improvement of SCADA System Used for Electrical Network Control by Incorporating Exceptional Handling Techniques, Proc. International Conf on Computer Applications in Electrical Engineering, University of Roorkee, pp. 703-709, 1977
2. Software Safety and Reliability Prediction to Achieve Fault Tolerance in Real Time Systems, Proc. of International Conf on Safety and Fire Engineering, CUSAT, pp. 120-128, 1999
3. Verification of Software in Machine Critical Applications, Proc. of International Conf on Safety and Fire Engineering, CUSAT, pp. 144-153, 1999
4. Cryptography, Proc of National Seminar on Business and Industry, Computer Society of India, pp. 38-44, 2000

In addition to these, following papers are published during the course of this work.

5. New Directions in Cryptography.
6. Role of Trap Door Functions in Mathematical Computations.
7. Secured Money Transactions Through Internet.
8. Fault Tolerant Computing Practices.
9. Pseudo Random Number Generation Facility in Various Computer Languages.
10. Real Time Computer Applications in Medical Science.
11. Involvement of Computers in Contemporary Socio Economic Transactions.

REFERENCES

- [ADA 1993] ADAMS C, TAVARES S E, Designing S Boxes for Ciphers Resistant to Differential Crypt Analysis. Pro. 3 rd Sym. on State and Progress of Research in Cryptography, pp. 181-190, 1993
- [AMI 1988] AMIRAZIZI H, HELLMAN M, Time Memory Processor Tradeoffs, IEEE Transactions on Information Theory, 34, pp. 505-512, 1988
- [AND 1995] ANDERSON R, On Fibonacci Key Stream Generators, Fast Software Encryption, Second Intl Workshop, pp. 346-352, 1995
- [AND 1996] ANDERSON R, BIHAM E, Two Practical and Provably Secure Block Ciphers : BEAR and LION, Fast Software Encryption, Third Intl Workshop, pp. 113-130, 1995
- [ANS 1981] ANSI X3.92, American National Standard- Data Encryption Algorithm, American National Standards Institute, 1981
- [ANS 1983] ANSI X3.106, American National Standard for Information Systems – Data Encryption Algorithm – Modes of Operation, ANSI, 1983
- [ANS 1985] ANSI X9.17, American National Standard – Financial Institution Key Management, ASCX9 , American Bankers Association, 1985
- [ANT 2001] Antonelli C J, Honeyman P, Wiretapping the Internet, Pro. of The International Society for Optical Engineering v 4232, pp. 75-84, 2001.
- [AOK 1996] AOKI K, OHTA K, Differential Linear Crypt Analysis of FEAL-8, IEICE Tran. on Fundmntls of Electronics, Commn and Computer Science, E79-A, pp. 20-27, 1996
- [APO 2000] Apostolopoulos T, Time Dependent Network Fault Management Model in a TCP/IP Environment, Computer Systems Science and Engg v 15 n 3, pp 165-174, 2000
- ATK 1997] Atkinson Randall J, Towards a More Secure Internet, Computer v 30, n 1, 1997
- [AVI 1971] Avizienis A The STAR : An Investigation Of The Theory And Practice Of F T C Design. IEEE Trans. On Computers C-20, 11, pp. 1312-1321, Nov 1971
- [AVI 1975] Avizienis A, Architecture of Fault Tolerant Computing Systems. Proc. of 5 th IEEE Symp on Fault-Tolerant Computing, pp. 3-16, June 1975
- [AVI 1977] Avizienis A, Fault Tolerant Computing – Progress, Problems and Prospects. Proc. of IFIP Congress on Information Processing, pp. 405 – 420, Aug, 1977
- [BAJ 2001] Bajpai G., Chang B C, Lau A, Reconfiguration of Flight Control Systems For Actuator Failures, IEEE Aerospace and Electronic Systems Magazine v 16 n 9, 2001

- [BAL 1999] Baldoni Roberto, Quaglia Francesco, Index-based Checkpointing Algorithm For Autonomous Distributed Systems, IEEE Tran. on Parallel and Distributed Systems v 10 n 2, pp 181-192,1999.
- [BAR 1977] BARKER W, Crypt Analysis of the Hagelin Cryptograph, Aegan Park Press, Laguna Hills, California, 1977
- [BEK 1982] BEKER H, PIPER F, Cipher Systems : The Protection of Communications, John Wiley & Sons, New York, 1982
- [BEN 1996] BEN AROYA I, BIHAM E, Differential Crypt Analysis of Lucifer, Advances in Cryptology, 9, pp. 21-34, 1996
- [BEN 1997] BENNET C,BRASSARD G, EKERT A, Quantum Cryptography, Scientific American, Special Issue, pp. 164-171, 1997
- [BEN 2001] Benantar M,The Internet Public Key Infrastructure, IBM Systems Journal v 40 n 3, pp. 648-665, 2001
- [BER 1968] BERLEKAMP E R, Factoring Polynomials Over Finite Fields, Bell System Technical journal, 46, PP.1853-1859, 1968
- [BER 1998] Berman Oded, Reliability Analysis of Communicating Recovery Blocks, IEEE Tran. on Reliability v 47 , pp 245-254, 1998
- [BER 2000] Berkovich Simon Y, Reversing the Error-Correction Scheme For A Fault-Tolerant Indexing, Computer Journal v 43 n 1, pp. 54-64, 2000.
- [BET 1995] Beth Thomas, Confidential communication on the Internet, Scientific American v 273 n 6, p 88-91, 1995.
- [BIH 1991] BIHAM E, SHAMIR A, Differential Crypt Analysis of FEAL and N-Hash, Advances in Cryptology- EUROCRYPT 91, pp. 1-16, 1991
- [BIH 1993] BIHAM E, SHAMIR A, Differential Crypt Analysis of Data Encryption Standard, Springer-Verlag, New York, 1993
- [BIH 1994] BIHAM E, New Types of Crypt Analytic Attacks Using Related Keys, Advances in Cryptology- EUROCRYPT 93, pp. 398-409, 1994
- [BIH 1994A] BIHAM E, New Types of Cryptanalytic Attacks Using Related Keys, Journal Of Cryptology, 7, pp. 229-246, 1994
- [BIH 1995] BIHAM E, Cryptanalysis of Multiple Mode of Operation, Advances in Cryptology – ASIACRYPT 94, pp.278-292, 1995
- [BIH 1995A] BIHAM E, BIRYUKOV A, How to Strengthen DES Using Existing Hardware, ASIACRYPT 95, pp.398-412,1995

- [BHA 1996] Bhat G Mohiuddin, Ahmad Waseem, Reliable & Secure Data Transmission, Electronic Engineering v 68 n 832, pp. 32-34, 1996
- [BIH 1992] BIHAM E, SHAMIR A, differential Crypt Analysis of Snefru, Khefre, REDOC II, LOKI and LUCIFER, CRYPTO 91, pp. 156-171, 1992
- [BLA 1987] BLAHUT R E, Principles and Practice of Information Theory, Addison-Wesley, 1987
- [BLA 1998] Blackburn S R, Cryptanalysis of Keystream Generator due to Chan and Cheng, Electronics Letters v 34 n 18, pp. 1737-1738, 1998.
- [BLO 1994] BLOCHER U, DICHTL M, Fish : A Fast Software Stream Cipher, Fast Software Encryption, Cambridge Security Work., pp.41-44, 1994
- [BOR 1974] Borgerson B.R. and Freitas R.F., 'PRIME' Using A New Reliability Model. Proc. of 4th IEEE Int. Symp. on FTC, pp. 26 -31, June 1974.
- [BOB 1994] BOBSHAW M J B, On Evaluating the Linear Complexity of a Sequences of Least period 2^n , Designs, Codes and Cryptography, 4, pp. 263-269, 1994
- [BRA 1988] BRANSTAD D K, Modern Cryptology : A Tutorial, Springer, New York, 1988
- [BRA 1996] Brasileiro Francisco V, Ezhilchelvan Paul Devadoss, Shrivastava Santosh K, Tao S, Implementing fail-silent nodes for distributed systems, IEEE Trans on Computers v 45 n 11, pp. 1226-1238, 1996
- [BRO 1990] BROWN L, PIEPRZYK J, SEBERRY J, LOKI- A Cryptographic Primitive for Authentication and Secrecy Applications. AUSCRYPT 90, pp. 229-236, 1990
- [BRO 1993] BROWN L, KWAN M, PIEPRZYK J, Improving Resistance to Differential Crypt Analysis and the Redesign of LOKI, ASIACRYPT 91, pp. 36-50, 1993
- [BRS 1988] BRASSARD G, Modern Cryptology : a Tutorial, LNCS 325, Springer-Verlag, 1988
- [BOU 1971] Bouricius, W.G. et al. Reliability Modeling for F T C. IEEE Trans. on Computers, C-20,11, pp. 1306-1311, Nov. 1971.
- [CAM 1993] CAMPBELL K W, WIENER M J, DES is not a group, Advances in Cryptology – CRYPTO 92, pp. 512-520, 1993
- [CAL 2000] Caloyannides Michael A, Encryption wars: Shifting tactics, IEEE Spectrum v 37 n 5, pp. 46-51, 2000.
- [CHA 1995] CHARNES C, O'CONNOR L, PIEPRZYK J, Comments on Soviet Encryption Algorithm, EUROCRYPT 94, pp.433-438, 1995
- [CHA 1998] Chang Chin Chen, Wu Tzong Chen, Broadcasting Secrets In Commn Networks, Computer Systems Science And Engg v 13 n 2, pp. 121- 127, 1998

- [CHA 1999A] Chau Savio N, Alkalai Leon, Tai Ann T, Design of a Fault-Tolerant COTS-Based Bus Architecture, IEEE Trans on Reliability v 48 n 4, pp. 351-359, 1999
- [CHA 2002] CHAN K C, CHAN S H C, Distributed Servers Approach For Large-Scale Secure Multicast, IEEE Journal on Commn. V 20(8), pp. 1500-1510, 2002.
- [CHE 1991] CHEPYZHOV V, SMEETS B, On a Fast Correlation Attack on Certain Stream Ciphers, EUROCRYPT 91, pp.176-185, 1991
- [CHE 1996] Cheng Jeffrey K, Ha Liuzhu, Encryption Method With Random Intervening Stripes, Proceedings of SPIE - The Intl Society for Optical Engg v 2885, pp. 11-16, 1996
- [CHE 1997] Chen Biao, Kamat Sanjay, Fault-Tolerant, Real-Time Communication In FDDI-Based Networks, Computer v 30 n 4, pp 83-90, 1997
- [CHI 1996] Chiu Ge Ming, Young Cheng Ru, Efficient Rollback-Recovery Technique In Distributed Computing Systems, IEEE Transactions on Parallel and Distributed Systems v 7 n 6 Jun 1996. p 565-577, 1996
- [CON 2000] CONSTANTINESCU CRISTIAN, Teraflops Supercomputer: Architecture And Validation Of The Fault Tolerance Mechanisms, IEEE Transactions on Computers v 49 n 9, pp. 886-894, 2000
- [COP 1986] COPPERSMITH D, The Real Reason for Rivest's phenomenon, Advances in Cryptology- CRYPTO 85, pp. 535-536, 1986.
- [COP 1994] COPPERSMITH D, KRAWCZYK H, MANSUR Y, The Shrinking Generator, Advances in Cryptology – CRYPTO 93, pp. 22-39,1994
- [COP 1994A] COPPERSMITH D, The DES and its Strength Against Attacks, IBM. Journal of Research and Development, v 38, pp. 243-250, 1994
- [COP 1995] COPPERSMITH D, ROGAWAY P, Software-Efficient Pseudorandom Function and the Use thereof for Encryption, US Patent #5, 454039, 26 Sept, 1995
- [COP 1996] Coppersmith D, Johnson D B, Matyas S M, Proposed Mode For Triple-DES Encryption, IBM Journal of Research and Devel. , v 40 n 2, pp. 253-262, 1996
- [COV 1978] COVER T, A Convergent Gambling Estimate of the Entropy of English, IEEE Tran on Information Theory, 24, pp. 413-421, 1978
- [DAE 1994] DAEMEN J, GOVAERTS R, A New approach to Block Cipher Design, Fast Software Encryption, Cambridge Security Workshop, pp. 18-32, 1994
- [DAE 1994A] DAEMON J, GOVAERTS R, VANDEVALLE J, Weak Keys for IDEA, Advances in Cryptology –CRYPTO 93, pp.224-231, 1994
- [DAE 1994B] DAEMEN J, GOVAERTS R, Resynchronisation Weakness in Synchronous Stream Ciphers, EUROCRYPT 93, pp. 159-167, 1994

- [DAE 1995] DAEMEN J, Cipher and Hash Function Design, Katholieke Universiteit Press, 1995
- [DAI 1991] DAI Z D, YANG J H, Linear Complexity Of Periodically Repeated Random Sequences, EUROCRYPT 91, pp.168-175, 1991
- [DAV 1983] DAVIES D W, PARKIN G I P, The Average Cycle Size of the Key Stream in Output Feedback Decipherment, Proc. of CRYPTO 82, pp. 97-98, 1983
- [DAV 1989] DAVIES D W, PRICE W L, Security for Computer Networks, John Wiley & Sons, New York, 2 nd Edition, 1989
- [DAW 1993] DAWSON E, Cryptanalysis of Summation Generator, Advances in Cryptology-AUSCRYPT 92, pp. 209-215, 1993
- [DEB 2001] Deb S, Domagala C, Ghoshal S, Patterson Hine A, Alena R, Remote Diagnosis Of The International Space Station Utilizing Telemetry Data, Pro of SPIE - Intl Society for Optical Engg v 4389, pp. 60-71, 2001
- [DEN 1976] Denning D.E, A Lattice Model of Secure Information Flow. Communications of the ACM, 19,5, PP. 236-243, May 1976.
- [DEN 1983] DENNING D E, Cryptography and Data Security, Addison Wesley, 1983
- [DIF 1976] DIFFIE W, HELLMAN M E, Multi User Cryptographic Techniques, Proc. Of AFIPS National Computer Conf., pp. 109-112, 1976
- [DIF 1976A] DIFFIE W, HELLMAN M E, New Dimensions in Cryptography, IEEE Transactions on Information Theory, 22, 644-654, 1976
- [DIF 1977] DIFFIE W, HELLMAN M E, Exhaustive Cryptanalysis of the NBS Data Encryption Standard, Computer , 10, pp. 74-84, 1977
- [DIF 1979] DIFFIE W, HELLMAN M E, Privacy and Authentication : Proceedings of the IEEE,67, pp. 397-427, 1979
- [DIF 1992] DIFFIE W, The First Ten Years of Public Key Cryptology, The Science of Information Integrity. IEEE Press, pp. 135-175, 1992
- [DOR 2001] Domaratsky Y, Ingulets A, Alkhovik A, Back-end software for highly dependable real-time control systems, Proc - IEEE Computer Society's Intl Computer Software and Applns Conf, pp. 237-244, 2001
- [DRE 1996] Drew Steven J, Fail-safety Techniques And Their Extensions To Concurrent Systems, Computer Languages v 22 n 4, pp. 193-203, 1996.
- [DUT 1997] Datta Ajoy Kumar; Thiagarajan Visalakshi, Simulation of Self - Stabilizing Sgorithms, Computer Systems Science and Engineering v 12 n 5, pp. 295-306, 1997

- [DUT 1997A] Dutt Shantanu, Mahapatra Nihar R, Node-Covering, Error-Correcting Codes And Multiprocessors With Very High Average Fault Tolerance, IEEE Trans on Computers v 9, pp. 997-1015, 1997
- [EBE 1993] EBERLE H, A High Speed DES implementation for Network Applications, Advances in Cryptology, CRYPTO 92, pp. 521-539, 1993
- [ELG 1985] ELGAMAL T, A Public Key Crypto System and a Signature Scheme based on Discrete Logarithms, IEEE Tran on Information Theory, 31,pp. 469-472, 1985
- [ESK 2001] Eskicioglu A M, A Key Transport Protocol Based On Secret Sharing An Application To Conditional Access Systems, Proc of SPIE - The Intl Society for Optical Engineering v 4314, pp. 139-148, 2001.
- [EVE 1985] EVEN S, GOLDREICH O, On the Power of Cascade Ciphers, ACM Transactions on Computer Systems, 3, pp. 108-116,1985
- [FEI 1973] FEISTEL H, Cryptography and Computer Privacy, Scientific American, v 228,pp. 15-23, May 1973
- [FEI 1988] FEISTEL H, NOTZ W A and SMITH J L, Some Cryptographic Techniques for Machine to Machine Data Commn, Pro. of the IEEE, 63, pp. 1545-1554, 1975
- [FEL 1990] FELDMIER D C, KARN P R, UNIX Password Security – Ten Years Later, Advances in Cryptology – CRYPTO 89 pp. 44- 63, 1990
- [FIA 1991] FIAT A, NAOR M, Rigorous Time / Space Tradeoffs for Inverting Functions. Proc. of the 23 rd annual ACM Symp. on Theory of Computing, pp. 534-541, 1991
- [FIP 1977] FIPS 46, DES, Federal Information Processing Standards Pub 46, U S Dept of Commerce / N B S, National Technical Information Service, 1977
- [FIP 1981] FIPS 81,DES Modes of Operation, Federal Inf. Processing Standards PublN 112, US Department of Commerce / N B S, National Technical Information Service,1981
- [FIP 1981A] FIPS 74, guidelines for implementing and Using the NBS Data Encryption Standard, Federal Information Processing Standards Publication 74, US Dept. of Commerce / N B S , National Technical Information Service, Springfield,1981
- [FIP 1994] FIPS 185, Escrowed Encryption Standard, US Dept. of Commerce / NIST National Technical Information Service, Virginia, 1994
- [FISH 1973] Fischler H.A, and Firschein O, A Fault-Tolerant Multiprocessor Architecture for Real-Time Control Applications. Proc. of IEEE Symp. on Computer Architecture, Florida, pp. 151-157, Dec. 1973.
- [FRI 1920] FRIEDMAN W, The Index of Coincidence and it's Appln in Cryptography, California, 1979, First published in 1920.

- [FRI 1923] FRIEDMAN W, Elements of Crypt Analysis, Aegan Park Press, Laguna Hills, California, 1976, First Published in 1923
- [FRI 1944] FRIEDMAN W, Cryptanalysis, US Government Printing Office, Washington , 1944
- [GAI 1956] GAINES H, Cryptanalysis : A Study of Ciphers and Their Solutions, Dover Publications, New York, 1956
- [GAM 1983] GAMES R A, CHAN A H, A Fast Algorithm for Determining the Complexity of a Binary Sequence with Period 2^n , IEEE Tran. on Info. Theory, 29, pp. 144-146, 1983
- [GEF 1973] GEFPE P, How to Protect Data with Ciphers that are Really Hard to Break, Electronics,46, pp. 99-101, 1973
- [GHO 1997] Ghosh Sunondo, Melhem Rami, Mosse Daniel, Fault-Tolerance Through Scheduling Of A Periodic Tasks In Hard Real-Time Multiprocessor Systems, IEEE Transactions on Parallel and Distributed Systems v 8 n 3, pp. 272-284, 1997
- [GIL 1991] GILBERT H, CHASSE G, Statistical Attack of the Feal-8 Cryptosystems, Advances in Cryptology-CRYPTO 90, pp. 22-33, 1991
- [GIL 1994] GILBERT H, CHAUVAUD P, A Chosen plain Text Attack of the 16 Round Khufu Cryptosystem, Advances in Cryptology - EUROCRYPT 94, pp. 359-368, 1994
- [GOL 1982] GOLOMB S W, Shift Register Sequences, Holden Day, san Francisco, 1982
- [GOL 1990] GOLDWASSER S, The Search for Provably Secure Cryptosystems, Cryptology and Computational Number Theory , American Mathematical Society, pp. 89 –113, 1990
- [GOL 1994] GOLIC J, On the Security of Shift Register Based Keystream Generators, Fast Software encryption, Cambridge Security Workshop, pp.90-100, 1994
- [GOR 1983] GORDON J, RETKIN H, Are Big S boxes Best?, Proceedings of the Workshop on Cryptography, pp. 257-262, 1983
- [GUN 1988] GUNTER C G, A Universal Algorithm for Homophonic Coding, Advances in Cryptology- EUROCRYPT 88, pp.405-414, 1988
- [HAI 1976] Haio D.K, and Baum R.I, Information Secure Systems. Advances in Computers, 14, Academic Press, pp. 231-272, 1976.
- [HAJ 2002] HAJICOSTICS, VERGHESE C G, F T C in Groups And Semigroups, Jnl of the Frankiin Institute. 339(4-5), pp.387-430, 2002.
- [HAM 1972] Hamer-Hodges K J, Fault Resistance and Recovery within System 250, Proc. of Int. Conf. On Computer Communications, pp. 290-296, 1972.
- [HAN 1998] Han Seungjae, Shin Kang G, Primary-Backup Channel Approach To Dependable Real-Time Communication In Multihop Networks, IEEE Trans on Comp v 47 n 1, pp. 46-61, 1998

- [HAR 1985] HARPEL C, KRAMER G G and MASSEY J L, A Generalisation of Linear Crypt Analysis, Advances in Cryptology – EUROCRYPT 95, pp. 24-38, 1995
- [HAS 1999] Hastad Johan, Levin Leonid A, Luby Michael, Pseudorandom Generator From Any One-Way Function, SIAM Journal On Computing V 28 N 4, Pp. 1364-1396, 1999
- [HEL 1976] HELLMAN M E, MERKLE R, Results of an Initial Attempt to Crypt Analyse the NBS DES, Technical Report, Information systems Laboratory, Stanford University, 1976
- [HEL 1977] HELLMAN M E, an Extension of Shannon Theory Approach to Cryptography, IEEE Transactions on Information theory, 23, pp. 289-294, 1977
- [HEL 1980] HELLMAN M E, A Cryptanalytic Time Memory Trade Off, IEEE Transactions on Information Theory, 26, pp. 401-406, 1980
- [HER 2000] Herman Ted, Phase clocks for transient fault repair, IEEE Transactions on Parallel and Distributed Systems v 11 n 10, pp. 1048-1057, 2000
- [HER 2001] Herzberg A, Kutten S, Early detection of message forwarding faults, SIAM Journal on Computing v 30 n 4, pp. 1169-1196, 2001
- [HIL 1929] HILL L S, Cryptography in an Algebraic Alphabet, American Mathematical Monthly, 36, 306-312, 1929
- [HIL 1996] Hiltunen Matti A, Schlichting Richard D, Adaptive distributed and fault-tolerant systems, Computer Systems Science and Engineering v 11 n 5, pp 275-285, 1996.
- [HOF 1977] HOFFMAN L J, Modern Methods for Computer Security and Privacy, Prentice Hall, Eaglewood, Cliffs, New Jersey, 1977
- [HOP 1975] Hopkins A L and Smith T B, The Architectural Elements of a Symmetric Fault-Tolerant Multiprocessor, IEEE Trans. On Computers, C-24,5, pp. 498-505, May 1975
- [HUA 2001] Huang Y, Garcia Molina H, Replicated condition monitoring, Proceedings of the Annual ACM Symposium on Principles of Distributed Computing, pp.229-237, 2001
- [ISO 1988] ISO 8732, Banking – Key Management, International Organisation for Standardisation, Geneva, Switzerland, 1988
- [ISO 1991] ISO / IEC 10116, Information Processing – Modes of Operation for an n- Bit Block Cipher Algorithm, International Organisation for Standardisation, Geneva, Switzerland, 1991
- [JAN 1990] JANSEN C J A, BOEKEE D E, The Shortest Feedback Shift Register that can Generate a Given Sequence, Advances in Cryptology-CRYPTO 89, pp. 90-99, 1990
- [JAN 1990A] JANSEN C J A, BOEKEE D E, On the Significance of the Directed Acyclic Word Graph in Cryptography, Advances in Cryptology-CRYPTO 89, pp. 90-99, 1990

- [JEN 1990] JENDAL H N, KUHN Y J B, and MASSEY J L, An Information-theoretic Treatment of Homophonic Substitution, EUROCRYPT 89, pp. 382 - 394, 1990
- [JUE 1983] JUENEMAN R R, Analysis of Certain Aspects of Output Feedback Mode, Advances in Cryptology – Proceedings of CRYPTO 82, pp. 99-127, 1983
- [KAH 1967] KAHN D, The Codebreakers, Macmillian Publishing Company, New York, 1967.
- [KAL 1995] KALISKI B S, YIN Y L, On Differential and Linear Crypt analysis of RC5 Encryption Algorithm, proc. of Advances in Cryptology- CRYPTO 95, pp.171-184, 1995
- [KAL 1988] KALISKI B S, RIVEST R L, SHERMAN A T, Is the Data Encryption Standard a Group, Journal of Cryptology, 1, pp. 3-36, 1988
- [KAM 1979] KAM J and DAVIDA G, Structured Design of Substitution-Permutation Encryption Networks, IEEE Transactions on computers, 28, pp. 747- 753, 1979
- [KAN 1996] Kang Sang Hyuk, Chung Min Young, Sung Dan Keun, Analysis Of Satellite On-Board Time-Space-Time Switching Networks With Multiple Separated Space Switches. IEEE Transactions on Reliability v 45 n 2, pp 316-320, 1996
- [KAR 2000] Karri Ramesh, Kim Kyosun, Potkonjak Miodrag, Computer Aided Design Of Fault-Tolerant Application Specific Programmable Processors, IEEE Transactions on Computers v 49 n 11, pp. 1272-1284, 2000.
- [KAS 2000] Kasselmann P R, Penzhorn W T, Cryptanalysis of Reduced Version of HAVAL, Electronics Letters v 36 n 1, pp. 30-31, 2000.
- [KER 1883] KERCHOFFS A, La Cryptographie Militaire, Journal des Science Militaires, 9th series, pp. 161- 191, 1883
- [KIL 1996] KILIAN J, ROGAWAY P, How to Protect DES Against Exhaustive Key Search, Advances in Cryptology- CRYPTO 96, pp. 252-267, 1996
- [KIM 1996] Kim Hagbae, Shin Kang G, Design And Analysis Of An Optimal Instruction-Retry Policy For TMR Controller Computers, IEEE Tran on Computers v 45 n 11, pp 1217-1225, 1996.
- [KLA 1994] KLAPPER A, GORESKY M, 2- Adic Shift Registers, Fast Software Encryption, Cambridge Security Workshop, Springer- Verlag, pp. 174-178, 1994
- [KLA 1994A] KLAPPER A, The Vulnerability of Geometric Sequences Based on Fields of Odd characteristic, Journal of Cryptology, 7, pp. 33-51, 1994
- [KLA 1995] KLAPPER A, GORESKY M, Cryptanalysis Based on 2 Adic Rational Approximation, advances in Cryptology-CRYPTO 95, pp.262-273, 1995
- [KLA 1996] KLAPPER A, GORESKY M, Feedback Shift Registers, Combines with Memory and 2 Adic Span, Journal of Cryptology, pp.112-119, 1996

- [KNU 1988] KNUTH D E, The Art of Computer Programming – Semi numerical Algorithms, Vol 2, Addison-Wesley, Reading, Massachusetts, 1981
- [KNU 1993] KNUDSEN L R, Crypt Analysis of LOKI, ASIACRYPT91, pp. 22-35, 1993
- [KNU 1994] KNUDSEN L R, Block Ciphers – Analysis, Design and Applications, Arhus University Publication, Denmark, 1994
- [KNU 1995] KNUDSEN L R, A Key- Schedule Weakness in SAFER K-64, Advances in Cryptology- CRYPTO 95, pp. 274-286, 1995
- [KNU 1996] KNUDSEN L R, BERSON T, Truncated Differentials of SAFER, Fast Software Encryption, Third International Workshop, Springer- Verlag, pp. 15-26, 1996
- [KNU 1996A] KNUDSEN L R, MEIER W, Improved Differential Attacks on RC5, Advances in Cryptology – CRYPTO 96, pp. 216-228, 1996
- [KOH 1978] KOHFELDER L M, A Method for Identification, MIT Laboratory for Computer Science pp. 39-43, 1978.
- [KON 1981] KONHEIM A G, Cryptography a Primer, John Wiley & Sons, New York, 1981
- [KOY 1993] KOYAMA K, TERADA R, How to Strengthen DES like Crypto Systems against Differential Crypt Analysis, IEEE Transactions on Fundamentals of Electronics, Communications and Computer Science, E76-A, pp. 63-69, 1993
- [KUS 1996] KUSUDA K, MATSUMOTO T, Optimisation of Time Memory Trade off Cryptanalysis and its Application to DES, FEAL-32, and Skijack, IEICE Transactions on Fundamentals of Electronics, Commn and Computer Science, pp. 35-48, 1996
- [LAN 1994] LANGFORD S K, HELLMAN M E, Differential Linear Crypt Analysis, Advances in Cryptology, Proc. Of CRYPTO 94, pp. 17-25, 1994
- [LAN 1997] Landwehr Carl E, Goldschlag David M, Security issues in Networks with Internet Access, Proceedings of the IEEE v 85 n 12, pp. 2034-2051, 1997
- [LAI 1991] LAI X, MASSEY J L, Aproposal for a New Encryption Standard, Advances in Cryptology, Proc, of EUROCRYPT 90, pp. 389-404, 1991.
- [LAI 1991A] LAI X, MASSEY J L, MURPHY S, Markov Model and Differential Crypt Analysis, Advances in Cryptology, Proc, of EUROCRYPT 90, pp. 17-38, 1991
- [LAI 1992] LAI X, On the Design and Security of Block Ciphers, ETH Series in Information Processing, Technische Hochschule, Zurich, 1992
- [LEE 2002] LEE H, MOON S, Parallel Stream Cipher For Secure High-Speed Communications, Signal Processing. 82(2), pp. 259-265, 2002.

- [LEE 2002A] LEE N Y, The Security Of The Improvement On The Generalization Of Threshold Signature And Authenticated Encryption, IEICE Transactions on Fundamentals of Electronics Communications & Computer Sciences. E85A(10), PP. 2364-2367, 2002.
- [LIA 1999] Liaw Horng Twu, Broadcasting cryptosystem in computer networks, Computers & Mathematics with Applications v 37 n 6, pp. 85-87, 1999.
- [LIH 1998] Lihao Xu, Bruck Jehoshua, Deterministic Voting In Distributed Systems Using Error-Correcting Codes, IEEE Transactions on Parallel and Distributed Systems v 9 n 8, pp 813-824, 1998.
- [LOC 1999] Lo Chi Chun, Chen Yu Jen, Secure communication mechanisms for GSM networks, IEEE Transactions on Consumer Electronics v 45 n 4, pp. 1074-1080, 1999
- [LUB 1988] LUBY M, RACKOFF C, How to Construct Pseudorandom Permutations from Pseudorandom functions, SIAM Journal on Computing, 17, pp. 373-386, 1988
- [LUB 1998] Lubaszewski Marcelo, Courtois Bernard, Reliable fail-safe system, IEEE Transactions on Computers v 47 n 2, pp. 236-241, 1998
- [LUC 1999] Lucks Stefan, Weis Ruediger, Hilt Volker, Fast encryption for set-top technologies, Proceedings of SPIE - The Intl Society for Optical Engg v 3654, pp. 84-94, 1999.
- [LYO 1962] Lyons R.E. and Vanderkulk W, The Use of Triple-Modular Redundancy to Improve Computer Reliability. IBM Journal of Res. and Dev., V 2, pp. 200-209, April 1962.
- [MAD 1984] MADRYGA W, a high Performance Encryption algorithm, Computer Security: A Global Challenge, Proc. of the Intl Conf on Computer Security, pp. 557-570, 1984
- [MAS 1969] MASSEY J L, Shift Register Synthesis and BCH Decoding, IEEE Transactions on Information Theory, 15, pp. 122-127, 1969
- [MAS 1985] MASSEY J L, INGEMARSSON I, The Rip Van Winkle Cipher – A simple and Provably Computationally Secure Cipher with a Finite Key, IEEE International Symposium on Information Theory, pp. 146, 1985
- [MAS 1992] MASSEY J L, Contemporary Cryptology : An Introduction, Contemporary Cryptology: The Science of Information Integrity, 1-39, IEEE Press, 1992
- [MAS 1995] MASSEY J L, SAFER K-64 : One Year Later, Fast Software Encryption, Intl. Workshop, Springer-Verlag, pp. 212-241, 1995
- [MAT 1970] Mathur F.P. and Avizienis A, Reliability Analysis and Architecture of a Hybrid-Redundant Digital System: Generalised Triple Modular Redundancy with Self-Repair. Proc. of AFIPS Spring Joint Computer Conf., 36, Atlantic City, pp. 375-383, 1970.
- [MAT 1975] Mathur P.P, and De Sousa P.T, Reliability Modeling and Analysis of General Modular Redundant Systems. IEEE Trans. Reliability, R-24, 12, pp. 296-299, Dec. 1975.

- [MAT 1993] MATSUI M, YAMAGISHI A, A New Method for Known Plain Text Attack of FEAL cipher, Proc. of Advances in Cryptology- EUROCRYPT 92, pp. 81-91, 1993
- [MAT 1995] MATSUI M, On Correlation Between the Order of S Boxes and the Strength of DES, Advances in Cryptology- CRYPTO 87, pp. 185-193, 1988
- [MAT 2000] Matsumoto Hiroyuki, Matsumoto Tsutomu, Evaluating Security Of A Clone Preventive Technique Using Physical Randomness And Cryptography, Proceedings of SPIE - The Intl Society for Optical Engineering v 3973, pp. 139-152, 2000.
- [MAU 1991] MAURER U, New Approaches to the Design of Self Synchronising Stream Ciphers, Advances in Cryptology- EUROCRYPT 91, pp. 458-471, 1991
- [MAU 1992] MAURER U, Conditionally Perfect and Secrecy And Provably Secure Randomized Cipher, Journal of Cryptology, 5, pp. 53-66, 1992
- [MAU 1993] MAURER U, Cascade Ciphers : The Importance of Being First, Journal of Cryptography, 6, pp. 55-61, 1993
- [MAU 1995] MAURER U, The Role of Information Theory in Cryptography. Codes and Ciphers : Cryptography and Coding IV, Institute of Mathematics and it's Applications, 1995
- [MAY 1994] MATSUI M, Linear Crypt Analysis Method for DES Cipher, Advances in Cryptology, proc. Of EUROCRYPT 93, pp.386-397, 1994
- [MAY 1994A] MATSUI M, The First Experimental Crypt Analysis of the Data Encryption Standard, Advances in Cryptology, proc. Of EUROCRYPT 94, pp.1-11, 1994
- [MCE 1987] MCELIECE R J, Finite Fields for Computer Scientists and Engineers, Kluwer Academic Publishers, Boston, 1987
- [MEI 1988] MEIER W, STAFFELBACH O, Fast Correlation Attack on Certain Stream Ciphers, journal of Cryptology, 1, pp.159-176, 1989
- [MEI 1991] MEIER W, STAFFELBACH O, Analysis of Pseudorandom Sequences Generated by Cellular Automata, Advances in cryptology- EUROCRYPT 91, pp.186-199, 1991
- [MEI 1992] MEIER W, STAFFELBACH O, Correlation Properties of Combiners With Memory in Stream Ciphers, Journal of Cryptology, 5, pp. 67-86, 1992
- [MEI 1994] MEIER W, On the Security of the IDEA Block Cipher, Advances in Cryptology – EUROCRYPT 93, pp. 371-385, 1994
- [MEI 1994A] MASSEY J L, SAFER k-64 : A Byte Oriented Block Ciphering Algorithm, Fast Software Encryption, Cambridge Security Workshop, pp. 1-17, 1994
- [MER 1976] Meraud C, Browaeys F, and Germain G, automatic Roll Back Techniques of the Copra Computer. Proc. of the 6 th IEEE Int. Sym on F T C, pp. 23-29, June 1976.

- [NEC 1992] NECHVATAL J, Public Key Cryptography, Contemporary Cryptography : The Science of Information Integrity, IEEE Press, pp. 177-288, 1992
- [NIC 1997] Nicolaidis Michael,Ricardo O. Manich Salvador, Figueras Joan, Fault-Secure Parity Prediction Arithmetic Operators, IEEE Design & Test of Computers v 14 n 2, pp. 60-71, 1997
- [OHT 1994] OHTA K, AOKI K, linear Crypt analysis of the Fast Data Encryption algorithm, Advances in Cryptology- CRYPTO 94, PP. 12-16, 1994
- [OOR 1991] OORSCHOT P V, WIENER M, A Known plain Text Attack on Two Key Triple Encryption, Advances in Cryptology -- EUROCRYPT 91, pp.210-218, 1991
- [OOR 1994] OORSCHOT P V, WIENER M, Improving Implementable Meet-In-The- Middle Attacks By Orders Of Magnitude, Advances in Cryptology - CRYPTO 96, pp. 229-236, 1996
- [PAT 2002] PATEL S, RAMZAN S, SUNDARAM G S, Security For Wireless Internet Access, Bell Labs Technical Journal. V.6(2), pp. 74-83, 2002.
- [PET 2000] Petrie Craig S, Connelly J Alvin, Noise-Based IC Random Number Generator For Applications In Cryptography, IEEE Transactions on Circuits and Systems : Fundamental Theory and Applications v 47 n 5, pp. 615-621, 2000
- [PFI 1993] PFITZMANN B, ASSMANN R, More Efficient Software Implementations of DES, Computers and Security,12, pp. 477-500, 1993
- [PLA 1997] Plank J. S, Tutorial on Reed-Solomon Coding For Fault-Tolerance in RAID-like systems, Software - Practice and Experience v 27 n 9, pp. 995-1012, 1997
- [POP 1975] Popek G.J. and Kline, C.S. A Verifiable Protection System, Proc. of Int. Conf. on Reliable Software, Los Angeles, pp. 291-304, April 1975.
- [PRE 1993] PRENEEL B, Analysis and Design of Cryptographic Hash Functions, Katholocke Universiteit Press, Belgium, 1993
- [PRE 1993A] PRENEEL B, Cryptographic Hash Functions, Kluwer Academic Publications, Boston,1993
- [PRE 1994] PRENEEL B, NUTTIN M and BULLENS J, Crypt Analysis of The CBF Mode of the DES with a Reduced Number of Rounds, CRYPTO 93, pp. 212-223, 1995
- [PRO 1985] PROCTOR N, A Self Synchronising Cascaded Cipher System with Dynamic Control of Error Propagation, Advances in Cryptology, Proc. of CRYPTO 84, 1985
- [RAB 1978] RABIN M O, Digital Signatures, Foundations of Secure Communication, Academic Press, pp.155-168, 1978
- [RAB 1979] RABIN M O, Digitalised Signatures and Public Key Functions as Intractable as Factorisation, MIT Laboratory for Computer Science,1979

- [RAY 1997] Ray Priest Michael S, Network Security Considerations In TCP/IP-Based Manufacturing Automation, ISA Transactions v 36 n 1, pp 37-48, 1997.
- [RIJ 1995] RIJMEN V, PRENEEL B, On Weakness of Non Surjective Round functions, Workshop on Selected Areas in Cryptography SAC 95, Ottawa, pp. 18-19, 1995
- [RIJ 1996] RIJMEN V, PRENEEL B, The Cipher SHARK, Fast Software Encryption, Third International Workshop, pp. 99-111, 1996
- [RIV 1978] RIVEST R L, SHAMIR A, ADLEMAN L M, A Method for Obtaining Digital Signatures and Public Key Crypto Systems, Comms. of the ACM, 21, pp. 120-126, 1978
- [RIV 1981] RIVEST R L, Statistical Analysis of the Hagelin Cryptograph, Cryptologia, 5, pp. 27-32, 1981
- [RIV 1983] RIVEST RL, SHERMAN A T, Randomised Encryption Techniques, Advances in Cryptology, Proc. of CRYPTO 82, pp.145-163, 1983
- [RIV 1990] RIVEST R L, Cryptography, Handbook of Theoretical Computer Science, 719-755, Elsevier Science Publishers, 1990
- [RIV 1995] RIVEST R L, The RC5 Encryption Algorithm, Fast Software Encryption, Second International Workshop, pp. 86-96, 1995
- [ROE 1995] ROE M, How to Reverse Engineer an EES Device, Fast Software Encryption, Second Intl. Workshop, pp. 3305-328, 1995
- [ROG 1994] ROGAWAY P, COPPERSMITH D, A Software Optimised Encryption Algorithm, Fast Software Encryption, Cambridge Security Workshop, pp. 56-63, 1994
- [ROM 1998] Romanovsky Alexander B, Predictable Toleration Of Design Faults: Recovery Blocks In Real Time Systems, Computer Science and Engg v 13 n 6, pp 369-377, 1998
- [ROS 2002] ROSA T, Encryption protection, IEEE Spectrum. V. 39(9), pp. 12-15, 2002.
- [RUB 1979] RUBIN F, Decrypting a Stream Cipher based on J K Flip flops, IEEE Transactions on computers, 28, pp. 483-487, 1979
- [RUE 1985] RUEPPEL R A, STAFFELBACH O J, Products of Linear Recurring Sequences with Maximum Complexity, IEEE Trans. on Information Theory, 33, pp. 124-131, 1985
- [RUE 1986] RUEPPEL R A, Analysis and Design of Stream Ciphers, Springer-Verlag, 1986
- [RUE 1992] RUEPPEL R A, Stream Ciphers, Contemporary Cryptology : The Science of Information integrity, IEEE Press, pp. 65-134, 1992
- [ROB 1995] ROBSHAW M J B, Stream Ciphers, Technical Report, TR-701, RSA Laboratories, 1995

- [SAL 1975] Saltier J.H, and Schroeder H.L, The Protection of Information in Computer Systems, Proceedings of the IEEE, 63,9,pp. 1278-1308, Sept. 1975.
- [SAL 1990] SLOMAA A, Public Key Cryptography, Springer-Verlag, Berlin, 1990
- [SAL 1998] Saleh Kassem, Al Saqabi Khaled, Error Detection And Diagnosis For Fault Tolerance In Distributed Systems, Information and Software Technology v 39 n 14, pp. 975-983 , 1998
- [SCH 1994] SCHNEIER B, Description of a New Variable Length Key, 64 Bit Block Cipher, Fast Software Encryption, Cambridge Security Workshop, pp. 191-204, 1994
- [SCH 1996] SCHNEIER B, Applied Cryptography : Protocols, Algorithms and Source Code in C, John Wiley & Sons, New York, 2 nd Edition, 1996
- [SCH 1997] Schollmeyer Martina, McMillin Bruce, General Method For Maximizing The Error-Detecting Ability Of Distributed Algorithms, IEEE Transactions on Parallel and Distributed Systems v 8 n 2, pp. 164-172, 1997
- [SEL 1966] SELMER E S, Linear Recurrence Relations Over Finite Fields, Dept. Of Mathematics, University of Bergen, Norway, 1966
- [SHA 1948] SHANNON C E, A Mathematical Survey of Communication, Bell System Technical Journal, 27, pp. 379-423, 1948
- [SHA 1949] SHANNON C E, Communication Theory of Secrecy Systems, Bell Systems Technical Journal, 28, pp. 656-715, 1949
- [SHA 1951] SHANNON C E, Prediction and Entropy of Printed English, Bell system Technical Journal, 30, pp. 50-64, 1951
- [SHE 1995] SHEPERD S, A High Speed Software Implementation of the Data Encryption Standard, Computers and Security,14, pp. 349-357, 1995
- [SHE 2002] SHEMTOOB D, SADOT D, System and method for information security in optical communication networks, Optical Engineering. 41(7), pp. 1621-1630, 2002.
- [SHI 1988] SHIMIZU A, MIYAGUCHI S, Fast Data Encipherment Algorithm FEAL, Advances in Cryptology – EUROCRYPT 87, pp. 267-278, 1988
- [SHO 2002] SHOUP V, GENNARO R, Securing Threshold Cryptosystems Against Chosen Ciphertext Attack, Journal of Cryptology. 15(2),pp.75-96, 2002.
- [SIE 1984] SIEGENTHALER T, Correlation Immunity of Non Linear Combining Functions for Cryptographic Applications, IEEE Trans on Information Theory, 30, pp.776-780, 1984
- [SIE 1985] SIEGENTHALER T, Decrypting a Class of Stream Ciphers Using Cipher Text Only Attack, IEEE Transactions on Computers, 34, pp. 81-85, 1985

- [SIM 1992] SIMMONS G J, A Survey of Information Authentication, Contemporary Cryptology : The Science of Information Integrity, pp. 379-419, 1992
- [SIM 1992A] SIMMONS C J, How to Insure Data Acquired to Verify Treaty Compliance are Trustworthy, Contemporary Cryptology : The Science of Information Integrity, IEEE Press, pp. 615-630, 1992
- [SIM 1997] Sims Terry, Real time recovery of fault tolerant processing elements, IEEE Aerospace and Electronic Systems Magazine v 12 n 12, pp. 13-18, 1997
- [SIN 1968] SINKOV A. Elementary Crypt Analysis : a Mathematical Approach, Random house, New York, 1968
- [SMI 1971] SMITH J L, The Design of Lucifer : A Cryptographic Device for Data Communications. I B M Research Report RC 3326, I B M T. J. Watson Research Centre, Yorktown Heights, New York, 1971
- [SMI 1992] SMID M E, BRANSTAD D K, The Encryption Standard Past and Future, Contemporary Cryptology : The Science of Information Integrity, IEEE Press pp. 43-64, 1992
- [SOM 1997] Somani Arun K, Vaidya Nitin H, Understanding Fault Tolerance And Reliability, Computer v 30 n 4, pp. 45-50, 1997.
- [SOR 1984] SORKIN A, Lucifer A Cryptographic Algorithm, Cryptologia, 8, pp. 22-35, 1984
- [SOU 2001] Soubusta J, Haderka O, Hendrych M, Quantum Random Number Generator, PThe International Society for Optical Engineering v 4356, pp. 54-60, 2001.
- [SRI 1999] Srinivasan Santhanam, Jha Niraj K, Safety and reliability driven task allocation in distributed systems, IEEE Transactions on Parallel and Distributed Systems v 10 n 3, pp 238-251, 1999
- [STI 1995] SSTINSON D R, Cryptography : Theory and Practice, CRC Press, Boca Raton, 1995
- [SUN 1997] Sun Hung Min, Private-key cryptosystem based on burst-error-correcting codes, Electronics Letters v 33 n 24, pp. 2035-3036, 1997
- [SUN 2002] SUN H M, Improving the information rate of a private-key cryptosystem based on product codes, Informatica. V.13(1), pp. 105-110, 2002.
- [SUR 2000] Surrey Peter J, Mucke Herzberg Dorothea, Security Of Remotely Operated Robotic Telescopes, Intl Society for Optical Engineering v 4011, pp. 145-156, 2000
- [SUS 2000] Susilo Willy, Safavi Naini Rei, Gysin Marc, Seberry Jennifer, New and Efficient Fail-Stop Signature Scheme, Computer Journal v 43 n 5 2000. p 430-437, 2000.
- [TAR 1992] TARDY CORFDIR A, GILBERT H, Aknown plain Text Attack of FEAL-4 and FEAL-6, Advances in Cryptology- CRYPTO 91, pp.172-182, 1992

- [TAK 1996] Takano Tadashi, Yamada Takahiro, Shutoh Kohshiro, Kanekawa Nobuyasu, In-Orbit Experiment On The Fault-Tolerant Space Computer Aboard The Satellite Hiten, IEEE Transactions on Reliability v 45 n 4, pp. 624-631, 1996.
- [TOK 1995] TOKITA T, SORIMACHI T, MATSUI M, On Applicability of Linear Crypt analysis to DES-like Crypto Systems – LOKI89, LOKI91 and S²DES, IEICE Trans on Electronics, Communications and Computer Science, E78-A, pp. 1148-1153, 1995
- [TSE 2002] TSENG Y M, JAN J K, CHIEN H Y, Digital Signature With Message Recovery Using Self-Certified Public Keys And Its Variants, Applied Mathematics & Computation. V.136(2-3) PP. 203-214, 2002.
- [TUC 1977] TUCHMAN W, Hellman Present no Shortcut Solutions to DES, IEEE Spectrum,16, pp. 40-41, 1979
- [VAN 1988] VAN TILBORG H C A, An Introduction to Cryptology, Kluwer Academic Publishers, Boston, 1988
- [VAU 1995] VAUDENAY S, On the Need for Multi Permutations : Cryptanalysis of MD4 and SAFER, Fast Software Encryption, Second Intl Workshop, pp. 286-297, 1995
- [VAU 1996] VAUDENAY S, On the Weak Keys of Blowfish, Fast Software Encryption, Second International Workshop, pp.27-32, 1996
- [VER 1926] VERNAM G S, Cipher Printing Telegraph Systems for Secret Wire and Radio Telegraphic Commn. Journal of The American Institute for Electrical Engineers, 1926
- [VOY 1985] VOYDOCKV L, KENT S T, Security Mechanisms in High Level Network Protocols, Computing Survey, 15, pp. 135-171, 1983
- [WAN 2001] Wang B, Sun C, Enhancement of Signal-To-Noise Ratio Of A Double Random Phase Encoding Encryption System, Optical Engineering v 40 n 8, pp. 1502-1506, 2001.
- [WAN 2002] WAANG Y G, A Comparison Of Two Approaches To Pseudo Randomness, Theoretical Computer Science. 276(1-2),pp.449-459, 2002.
- [WAY 1993] WAYNER P C, Content Addressable Search Engines and DES like Systems, Advances in Cryptology- CRYPTO 92, pp. 575-586, 1993
- [WEB 1986] WEBSTER A F, TAVEARES S E, On The Design of S Boxes, Advances in Cryptology – CRYPTO 85 pp. 523-534, 1986
- [WEI 1993] WIENER M J, Efficient DES Key Search, Technical Report TR –244, School of Computer Sciences, Carleton University, Ottawa,1993
- [WEN 1974] Wensley J H, Levitt K N and Neuman P G, A Comparative Study of Architectures for Fault-Tolerance. Proc. of 4 th IEEE Int. Symp. On FTC, pp. 116 –21, June 1974
- [WHE 1994] WHEELER D J, A Bulk Data Encryption Algorithm, Fast Software Encryption, Cambridge Security Workshop, pp. 127-134, 1994

- [WHE 1995] WHEELER D J, TEA: A Tiny Encryption Algorithm, Fast Software Encryption, Second Intl. Workshop, pp. 363-366, 1995
- [WIL 1972] Wilkes H.V, Time Sharing Computing Systems. American Elsevier / Macdonaid 1972.
- [WIL 1975] WILKES M V, Time Sharing Computer Systems, American Elsevier Pub Co, 1975
- [WOL 1986] WOLFRAM S, Cryptography with Cellular Automata, Advances in Cryptology-CRYPTO 85, pp. 429- 432, 1986
- [WRI 2002] WRIGHR D, Comparative Evaluation Of Electronic Payment Systems, Infor. V.40(1), pp.71-85, 2002.
- [WUC 2001] Wu C P, Jay Kuo C C, Fast Encryption Methods For Audiovisual Data Confidentiality, The International Society for Optical Engineering v 4209, pp. 284-295, 2001.
- [WUN 2000] Wu N Eva, Zhou Kemin, Salomon Gregory, Control Reconfigurability Of Linear Time-Invariant Systems, Automatica v 36 n 11, pp. 1767-1771, 2000
- [WUN 2001] Wunnava S V, Lule E, Distributed security schemes for networks, Conference Proceedings - IEEE SOUTHEASTCON, pp. 114-117, 2001.
- [XUJ 2000] Xu Jie, Romanovsky Alexander, Randell Brian, Concurrent Exception Handling And Resolution In Distributed Object Systems, IEEE Transactions on Parallel and Distributed Systems v 11 n 10, pp. 1019-1032, 2000.
- [YAN 1996] Yang Lieliang, Li Chengshu, Nie Tao, Fault-Tolerant Data Transmission Model Based On Redundant Residue Number System, Proo. f Conf, Intl Society for Optical Engineering . pp. 517-522, 1996.
- [YEN 2001] Yen G G., Ho L W, On-Line Intelligent Fault Tolerant Control For Catastrophic System Failures, Intl Society for Optical Engineering v 4, n 9, pp. 35-46, 2001
- [YIS 2001] Yi S Y, Ryu C S, Ryu D H, Lee S H, Evaluation Of Correlation In Optical Encryption Using Visual Cryptography, Intl Society for Optical Engg v 4387 pp. 238-246, 2001
- [YIX 2001] Yi X, Tan C H, Siew C K, Syed M R, Fast encryption for multimedia, IEEE Transactions on Consumer Electronics v 47 n 1, pp. 101-107, 2001.
- [YON 2000] Yonsei, Univ. Seoul, S Korea, Evaluation Of Fault Tolerance Latency From Real-Time Application's Perspectives, IEEE Trans on Computers v 49 n 1, pp. 55-64,2000
- [YUA 1997] Yuan Shyan-Ming; Agrawala Ashok K, Efficient Communication Structure For Decentralized Algorithms With Fault Tolerance, Computer Systems Science and Engineering v 12 n 6, pp 343-349, 1997
- [ZAL 2000] Zalevsky Zeev, Mendlovic David, Levy Uriel, Shabtay Gal, New Optical Random Coding Technique For Security Systems, Optics Comm. v 180 n 1, pp. 15-20, 2000.



- [ZEN 1990] ZENG K, HUANG M, On the Linear Syndrome Methods in Crypt Analysis. CRYPTO 88, pp.469-478, 1990
- [ZEN 1991] ZENG K, YANG C H, RAO T R N, An Improved Linear Syndrome Algorithm in Cryptanalysis with Applications, CRYPTO 90, pp. 34-47, 1991
- [ZHE 1998] Zheng Qin, Shin Kang G, Fault-tolerant real-time communication in distributed computing systems, IEEE Transactions on Parallel and Distributed Systems v 9 n 5, pp 470-480, 1998.
- [ZIV 1997] Ziv Av, Bruck Jehoshua, On-Line Algorithm For Checkpoint Placement, IEEE Transactions on Computers v 9, pp. 976-985, 1997
- [ZHO 1998] Zhou D H, Frank P M, Fault Diagnostics And Fault Tolerant Control, IEEE Transactions on Aerospace and Electronic Systems v 34 n 2, pp. 420-427, 1998