# International Journal of Computer Science and Security (IJCSS)

ISSN : 1985-1553

VOLUME 4, ISSUE 4

PUBLICATION FREQUENCY: 6 ISSUES PER YEAR

# International Journal of Computer Science and Security (IJCSS)

Volume 4, Issue 4, 2010

Edited By
Computer Science Journals
www.cscjournals.org

**Editor in Chief Dr. Haralambos Mouratidis**

# International Journal of Computer Science and Security (IJCSS)

Book: 2010 Volume 4, Issue 4
Publishing Date: 30-10-2010
Proceedings
ISSN (Online): 1985-1553



IJCSS Journal is a part of CSC Publishers
http://www.cscjournals.org


Published in Malaysia

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

**CSC Publishers**

# Editorial Preface

This is fourth issue of volume four of the International Journal of Computer Science and Security (IJCSS). IJCSS is an International refereed journal for publication of current research in computer science and computer security technologies. IJCSS publishes research papers dealing primarily with the technological aspects of computer science in general and computer security in particular. Publications of IJCSS are beneficial for researchers, academics, scholars, advanced students, practitioners, and those seeking an update on current experience, state of the art research theories and future prospects in relation to computer science in general but specific to computer security studies. Some important topics cover by IJCSS are databases, electronic commerce, multimedia, bioinformatics, signal processing, image processing, access control, computer security, cryptography, communications and data security, etc.

This journal publishes new dissertations and state of the art research to target its readership that not only includes researchers, industrialists and scientist but also advanced students and practitioners. The aim of IJCSS is to publish research which is not only technically proficient, but contains innovation or information for our international readers. In order to position IJCSS as one of the top International journal in computer science and security, a group of highly valuable and senior International scholars are serving its Editorial Board who ensures that each issue must publish qualitative research articles from International research communities relevant to Computer science and security fields.

IJCSS editors understand that how much it is important for authors and researchers to have their work published with a minimum delay after submission of their papers. They also strongly believe that the direct communication between the editors and authors are important for the welfare, quality and wellbeing of the Journal and its readers. Therefore, all activities from paper submission to paper publication are controlled through electronic systems that include electronic submission, editorial panel and review system that ensures rapid decision with least delays in the publication processes.

To build its international reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJCSS. We would like to remind you that the success of our journal depends directly on the number of quality articles submitted for review. Accordingly, we would like to request your participation by submitting quality manuscripts for review and encouraging your colleagues to submit quality manuscripts for review. One of the great benefits we can provide to our prospective authors is the mentoring nature of our review process. IJCSS provides authors with high quality, helpful reviews that are shaped to assist authors in improving their manuscripts.

**Editorial Board Members**
International Journal of Computer Science & Security (IJCSS)

# Editorial Board

## Editor-in-Chief (EiC)



## Associate Editors (AEiCs)



## Editorial Board Members (EBMs)

# Table of Content

Volume 4, Issue 4, October 2010

## Pages

# Deploying E-Mail as an Official Communication Tool: Technical Prospect

**Wasan Shaker Awad** Wasan_shaker@itc.uob.bh
*Department of Information Systems*
*Information Technology College*
*University of Bahrain*
*Sakheer, Bahrain*

**Abstract**

Applications are spreading widely through our lives entering every field controlling some of it and enhancing other; electronic mail, or e-mail, is the best known and most popular application of the Internet. More and more people see e-mail as a way to communicate quickly and affordably. Electronic mail can also provide an advanced means of communication and enhance the recently applied e-government program. The aim of this work is to study technically the official use of e-mail for the communication between the government and citizens. This paper is mainly for proposing an e-mail exchange system that can be used to fulfill the legal requirements, and to make the usage of e-mail as official communication tool is feasible.

**Keywords:** E-mail, E-government, Information Security, ICT.

## 1. INTRODUCTION

Mankind has always had a compelling desire to communicate, that's why many evolving communication methods had emerged through the years. Communication has revolutionized from traditional means to more advanced electronic communications. As a traditional way of communication, postal mail systems and courier services are considered to be the oldest forms of mail item distribution. Wherein written letters, and also small packages, are delivered to destinations around the world. Then communication has evolved to a new concept which is telecommunications, also known as electronic communication. Telecommunication is the transmission of data and information between computers using a communication link. It began in 1844, when Samuel Morse invented the telegraph, whereby sounds were translated manually into words using Morse code. Then in 1876, Alexander Graham Bell developed the telephone, which brought telecommunication into the home, and became crucial for business life for many years. After that facsimile (fax) was developed in the 1900s, which transmits documents over telephone lines. Technology continued to expand its influence, and another technology evolved in the telecommunication field. This technology was the Internet. Internet was mainly developed for the purpose of communication. One of the very first communication means through the Internet was e-mail [1].

In general, the official communication is done through traditional ways: sending the official letters or documents by post mail, and occasionally done by contacting the person by telephone or face-to-face. The message must be delivered by the correspondent person who works in the postal office to its destination. Then the receiver should approve the receipt of the message. To cope with the modern e-application revolution, there is a need to replace the traditional method of communication with more advanced and reliable electronic communication. The electronic mail, or e-mail system are best known and most popular network-based application. It is a way to communicate quickly and affordably. From here the idea of using the e-mail as official communication has emerged.

Adopting e-mail as official communication is a critical topic. The importance of this topic emerges from the wide use of e-applications, and the rapid improvements in the e-government filed. Hopefully the study results provide some small but valuable contribution to the research quiver especially to what is related to e-government field, and can consider as a trigger for a great revolution in the official communication.

The following points are considered as reasons that's motivates using electronic communication, and stimulate the need for an official E-mail Exchange System:

- The implementation of the e-government portal.
- The need of an electronic communication means for the fast, reliable and convenient communication that also copes with the emerging e-applications.
- The need of a new mailing system that can replace the current postal system and supports the extensive official communication securely and efficiently.
- The role of the modern technologies particularly the e-mail in enhancing the communication between the government and citizens.
- The inadequacy of current electronic communication infrastructure for the exchange of the sensitive, legal and private official communication.

Communication with both citizens and organizations involves the transmission of sensitive information and legal bindings. Consequently, electronic communication has to be highly secured. Although there is an infrastructure developed for commercial use of e-mails, this infrastructure is not securely sufficient for the exchange of the sensitive, legal and private official communication. As a result, there is a need of developing a new e-mail exchange system that fulfills the security and legal requirements. Thus, the main objective of this study is to investigate the feasibility of using e-mail as official communications, during the attempt to achieve the study's main objective many sub objectives will be delivered as well, and these objectives are:

- Provide a detailed study about the e-mail technology, its advantages, disadvantages, review of same case studies that has applied e-mail as official communication.
- Investigate the ability to implement this project technically.
- Provide a proposed system that supports the technical requirements of applying e-mail effectively and efficiently.

This paper is mainly for proposing an e-mail exchange system that can be used to fulfill the legal requirements, and to make the usage of e-mail is feasible.

## 2. LITERATURE REVIEW

E-mail, is short for electronic mail, is just an electronic message sent from one computer to another [2]. It's hard to remember what people's lives were like without e-mail. Ranking up there with the Web as one of the most useful features of the Internet, e-mail has become one of today's standard and preferred means of communication. E-mail usage by businesses became common several years before individuals began using it at home. Yet by the late 1990s, approximately 40% of all American householders owned a computer, and 26% of those families had Internet access. An analysis in 1998 indicated that there were 25 million e-mail users sending 15 billion messages per year [3].

E-mail is no longer just a method of communicating in business; it's a way of doing business. It has become an integral part of workers' lives. Most employees have Internet access at their work, and an e-mail account to help them in collaborating with their colleagues and customers, in order to be more productive at their work. A portion of those e-mails probably replaces the telephone calls or faxes or traditional mails [4]. For example, Western Provident Association (WPA) [5] which is one of Britain's leading health insurers, who insure over 500,000 people and more than 5,000 companies.

The main reason why people connect to the Internet is to communicate by e-mail. Traditional methods of communication are now converging onto the Internet – text messages, phone calls and video images can all be sent via the Internet. Furthermore, the growth in mobile communication and the continuing improvements in mobile communication devices means that e-mail is now accessible just about anywhere. So why e-mail taking the communication world by storm? These are the main benefits [6]: It is relatively low cost, easy to copy a message to many people at the same time, easy to

distribute information such as reports, spreadsheets, presentations and other files, personalized, convenient, relatively quick, and it does not sleep.

E-mail is a great tool used to communicate with others, however, with the added great advantages of e-mail, also comes some disadvantages such as: Viruses, Spamming, Flaming, Phishing, and E-mail Privacy and Security, such that, without some security protection can be compromised because [6]:

- E-mail messages are generally not encrypted.
- E-mail messages have to go through intermediate computers before reaching their destination, meaning it is relatively easy for others to intercept and read messages.
- In some business, e-mail messages of employees are monitored.
- Many Internet Service Providers (ISP) store copies of senders e-mail messages on their mail servers before they are delivered. The backups of these can remain up to several months on their server, even if the receiver deletes them in his/her mailbox.
- The received headers and other information in the e-mail can often identify the sender, preventing anonymous communication.

Although e-mail has several disadvantages, most of these can be solved easily. For example using an anti virus program provides protection from e-mail viruses, or some web mails provide message scanning for viruses. Spamming and phishing can also be handled by some web mail services. Moreover, most web mail services warn the user before opening an anonymous mail. Furthermore, there must be new rules and regulations to govern the use of Internet services, and to organize the exchange of the information across the world wide information net, mainly those information that are related to the official treatment and correspondences.

The problem of deploying email as official communication tool has been considered by a number of organizations and researchers. On November 6 and 7, 1997, with the sponsorship of the Markle Foundation, RAND convened a workshop in Washington, D.C. to begin a discussion of the character of the required infrastructure, who might plausibly provide it, how it might be financed, and what other policy changes (institutional, legal, programmatic) might be necessary to support secure communication between government and citizens. Attendees at the workshop included managers, policymakers, and analysts from a variety of government agencies at the state and federal levels and representatives of private-sector concerns that are users or providers, current or potential, of secure communications services [7].

At the Summit on the 21st Century Workforce, held June 20, 2001, in Washington, D.C., Secretary Elaine L. Chao announced the creation of a compliance E-Mail Initiative to ensure that the Department of Labor answers on a timely basis all electronic inquiries from DOL customers. This policy required all DOL agencies to establish and advertise electronic public contact mechanisms on DOL public Websites for collecting general comments, suggestions, or inquiries from the public and to develop procedures for handling electronic correspondence in accordance with this policy. This initiative provides the option for agencies to integrate electronic correspondence procedures with existing paper correspondence procedures [8]. The main purpose of this step is the establishment of OSHA E-Correspondence system, which provides for processing, routing, tracking, and responding to the public on general safety and health questions received through OSHA's public Website. The scope of this project was OSHA-wide. The project was based on the following basics:

- Department of Labor Electronic Correspondence Policy.
- Secretary's Order 2-2000, Department of Labor Internet Services, Section 6a.
- OSHA Instruction of Internet/Intranet Policy & Procedures of December 15, 2000.
- OSHA Instruction of Policy Issuances of December 11, 2000.
- OSHA Instruction of Non-Policy Issuances of December 11, 2000.

In order to achieve the above purpose the state must:

- Develop a system that ensures timely response to inquiries received through the OSHA E-Correspondence system.
- Notify OSHA through the Directorate of Cooperative and State Programs of any changes to the e-mail address designated to receive electronic correspondence.
- Maintain records of correspondence received and their responses to that correspondence.

The main offices and agencies that are involved in this project are National Office, Regional Offices, Area Offices, and State Plan States. These offices and agencies must implement the policies and

procedures contained in this project in order to ensure the consistency of the Correspondence. In addition to USA, Perry presets the use of e-mail in different countries [9]. Also, Yayehyirad [10] studied the possibilities offered by E-Government to Africa by documenting few initiatives on the continent that have developed innovative models that contribute to governments' efficiency, accessibility, transparency and accountability through the implementation of ICT based services. He also presented an application to provide a secure online email service to high level government officials. This implied the design and deployment of a corporate mail system for the government including the delivery and setup of mail servers.

This paper differs from previous studies by providing a general framework for deploying e-mail as an official communication tool between citizens. Although they presented an infrastructure for commercial use of e-mails, this infrastructure is not securely sufficient for the exchange of the sensitive, legal and private official communication. Consequently, developing a new e-mail exchange system that fulfills the security and legal requirements is needed.

## 3. THE OFFICIAL E-MAIL EXCHANGE SYSTEM LEGAL REQUIREMENTS

Deploying e-mails for official communications will be successful if:

- A number of legal rules and regulations should be set in order to govern the use of e-mail and the other e-communication means, and to organize the exchange of the information across the world wide information net.

- There is an awareness of the legislation interventions – locally and globally- to deal with the illegal and unlawful behaviors that it's performed via the e-mail and the other e-communication ways. In addition, there must be appropriate penalties against the individuals who cause those behaviors.

- There must be constraints that restrict the way of using e-mail in the official communication. The main purpose of these constraints is the assurance of the accuracy and the reliability of information being exchanged through such way of communication, at the same time these constraints are used to prevent the misuse of the citizen's sensitive information. Consequently, if these constraints were skipped there will be a stiff government penalties as well as civil suits. The responsibility in the e-communication field dose not confines on the management responsibility, but it includes the civil responsibility as well as the malefaction responsibility.

    a) **The Management Responsibility**: the management responsibility represented in the various penalties that usually issued by the management parties which has the role of mentoring and supervising the other parties that work in the E-communication field.

    b) **The Civil Responsibility**: regarding to the e-communication filed the civil responsibility is basically means the adherence of the individual citizens and the other parties that are involved in such communication to compensate the damages and harms they caused.

    c) **The Criminal Responsibility**: criminal malefaction responsibility is the inculpation of some of the actions that are related to the field of collecting, processing and distributing data.

- There must be continuance supervision and monitoring activities on the use of e-mails as the official communication, along with the ability of deciding on the legal responsibilities in case of skipping one or more of the rules and constraints that governs the use of e-mail as an official communication.

- There must be mechanisms to confirm that the message originated from its originator and that can take the place of the hand written signature usually used in the paper based official correspondent.

- There must be mechanisms for protecting the e-mail message that contains sensitive information, and to protect the behalf of the sender and the receiver, the sender and the receiver must agree upon a mechanism to confirm the reception of the e-mail message by its appropriate recipient [11].

## 4. THE OFFICIAL E-MAIL EXCHANGE SYSTEM SECURITY TECHNIQUES

The satisfaction of the above requirement requires the use of the following security techniques:

- **Message Encryption:** Protects the privacy of the message by converting it from plain, readable text into cipher (scrambled) text [12, 13].

- **Digital Signature:** An electronic, encryption-based, secure stamp of authentication on a message. The signature confirms that the message originated from the signer and has not been altered. Digital Signature provides Authentication, Non- repudiation and Data Integrity [14, 15].

- **Digital Certificate**: A digital means of proving your identity, using a public and private key pair. The private key is the secret part kept on the sender's computer that the sender uses to digitally sign messages to recipients and to decrypt (unlock) messages from recipients. Private keys should be password protected. The public key is sent to others or published in a directory, so that others can use it to send you encrypted messages. Mainly there the system will use two types of digital certificate: Identity and Authority Certificate [16].

    a) **Identity Certificate:** is the process of associating a public key with a particular user and establishing his identity.

    b) **Authority Certificate:** is the process of granting the user whose can now be verified, *authority* to access information, to make use of services, to carry out transactions, or whatever.

  Although this two functions (establishing identity Certificate and establishing authority Certificate) are distinct and quite separable and usually are initiated by different entities, in some cases they are initiated by the same entity, a government agency for example.

- **Conformation Mechanism:** An e-mail not reaching their destination is a growing problem, in case of the official communication this problem become more serious. Therefore there must be some mechanisms that can help both senders and receivers of emails to make sure their official emails are not lost. There are several confirmation mechanisms that can be used to ensure Official E-mail Exchange System delivery capabilities [17, 18]:

    a) **First: User Generated Feedback** is used to determine if Email is not flowing through the system. If no feedback is received then the system is presumed to be operating normally.

    b) **Second: Test Message Monitoring** is a variation of the user based feedback method. Instead of depending on user to notice non-delivery of E-mail messages, an administrator will periodically send a message through the E-mail system to a testing account. If the message is successfully delivered, the administrator assumes that the system is functioning properly.

    c) **Third: Looped Message Monitoring** is a variation of the test message monitoring method. An administrator still manually generates a message to test the E-mail system. The message is used to test the system by sending it through a loop to more than one testing accounts in different regions. If the test messages are successfully delivered, the administrator assumes that the system is functioning properly.

d) **Fourth: Automated Looped Message Monitoring** is a variation of the Looped Message monitoring method. Test message generation is automated by third party software. If the test messages are successfully delivered, the administrator assumes that the system is functioning properly.

e) **Fifth: Tracking Method** is based on using a tracking method such as the Pixel Tags. Pixel Tags are tiny invisible graphics (or minute embedded images) tucked away in HTML content distributed via e-mail that contain a set of instructions. When HTML-enabled e-mail clients open the HTML content, the pixel tag is instructed to contact a particular web server to receive a unique identifier code. This code is added to a special server log that records details of the machine and user receiving and opening the message. If the HTML content is forwarded to another HTML-enabled e-mail client, the pixel tag will perform similar functions, although it is limited in its ability to provide information on the referring machine. The information sent to servers can also be done by Web Bugs. Web Bugs are like the pixel tags described above. The affected e-mail clients at this point include Outlook 2000, Outlook Express, and Netscape 6 Mail Messenger, or any client which has JavaScript-functionality turned on by default.

f) **Sixth: Confirmation Software** using some of the confirmation software such as *Mailinfo* is another confirmation mechanism**.** Mailinfo allows the senders of emails to verify that the email messages have actually been received and notifies him/her the message has been read.

## 5. THE OFFICIAL E-MAIL EXCHANGE SYSTEM

This section suggests an official e-mail exchange system, which is a proposed solution to address all the necessary requirements in an official mailing system. The use of e-mail instead of the current postal system requires an e-mail exchange system with the following technical requirements:

- **Confidentiality:** Confidentiality means keeping information protected from unauthorized party.
- **Data Integrity:** Massages data is protected from unauthorized changes.
- **Authentication:** The process of identifying an individual**.** Citizens and government agencies must be sure that they are in fact communicating with the intended party
- **Non- repudiation:** A proof that a transaction occurred, or that user sent or received a message.
- **Access controls:** They are predicated on a system of identification and authentication -- that is, "Who are you?" and "Can you prove it?"
- **Confirmation of receiving and reading messages**.

Thus, the official E-mail exchange system can be implemented using a framework comprises three main technologies. These technologies are:

- Exchange server.
- E-mail client.
- (PKI ) Public key infrastructure.

These technologies are interacting together to provide a secure environment for the exchange of the official communications [16]. Each of these technologies is used to meet some of the official e-mail exchange system requirements and to implement one or more of the security services introduced previously.

- **Exchange Server.** Exchange server is messaging and collaborative software. Exchange's many features consist of electronic mail, calendaring, contacts and tasks, and support for the mobile and web-based access to information, as well as supporting data storage. Exchange server act as an access point for sending and receiving messages. The e-mail client should be accommodated with one or more of the previously mentioned confirmation mechanism.

- **E-mail Client.** E-mail client is a front-end computer program used to manage email. The E-mail client should be accommodated with software confirmation mechanism.

- **Public key infrastructure (PKI).** Public key infrastructure is an arrangement that used to manage keys and certificates. In such systems, each user has one or more key pairs, each comprising a "public" key that is known to his or her correspondents, and a "private" key known only to the user. These keys can be used as encryption keys and as signing keys.

At the heart of the PKI there should be one or more Certificate Authorities (CAs) also known as trusted third party (TTP). Certificate Authorities are trusted institutions or organizations that will mainly certify that a particular public key is associated with a particular user. Usually, Some Certificate Authorities (CAs) will make use of the S/MIME (Multipurpose Internet Mail Extensions) which is a standard for public key encryption and signing of e-mail to perform the following functions:

- The establishment of identity certificates.
- The use of the public key information to encrypt messages.
- The use of the public key information and the standard to verify the digital signature of a message, which was made using the signer's private key.
- For others to have confidence in this identity, a CA must also be able to provide nearly instantaneous verification that a particular user/public key pairing is still valid (that the user or other authority has not for some reason canceled a public key).
- CA will also provide customer services such as replacing certificates that have been lost or compromised, publishing directories of public keys, and assisting users who experience difficulties.

Other Certificate Authorities that may or may be the government represented by the Central Informatics Organization will use the same S/MIME standard for establishing the authority certificates by associating each electronic identity with specific records or in our case specific e-mail account. Note that the two functions of (establishing identity Certificate and establishing authority Certificate) can be performed by the same Certificate Authorities in some cases.

## 6. WHO CAN ACT AS CERTIFICATE AUTHORITIES FOR THE GOVERNMENT AGENCIES?

The previous sections considered the definition and the main functionalities of the Certificate Authorities, and concentrate on their role in the official e-mail exchange system. But the question now is "Who Can Act as Certificate Authorities for the Government Agencies?" Certificate Authorities can be any government agency or commercial institution that can meet the following criteria [7]:

- **Highly reliable identification of agencies and users.** The official communications usually include the transmission of extremely sensitive information. Government agencies and citizens will require a very high degree of confidence that they are in fact each communicating with the intended party.

- **Local Presence**. To ensure reliable identification of users, CAs may require in-person interactions and perhaps the physical presentation of certain documents. This in- person interaction may have to be repeated periodically to maintain the validity of the digital certificate. If secure electronic communication is to be available to any citizen who desires it, then every citizen will have to have easy access to an office of a suitable CA.

- **Extensive customer service.** Official E-mail Exchange System requires a robust customer service operation **(**to answer questions, to guide infrequent and perhaps unsophisticated users, and to restore or to replace lost or compromised certificates as examples).

Examples of the commercial institution that may be positioned to provide CA services for secure official communication are:

- **Specialist Firms have begun to offer CA services.** This kind of firms is established to serve relatively small and specialized populations**,** so they expand their operations to the entire population**.**

- **Banks.** Banks have ongoing trusted relationships with their customers and already go to some lengths to establish customers' identities. Banks have many points of presence in almost all

communities and, at least occasionally, deal face-to-face with their customers. Finally, many banks are moving toward creating electronic banking systems to serve their own customers. It may turn out that such bank infrastructures can be exploited for communications with the government at minimal additional cost**.**

- **Other institutions that maintain continuing relationships with individual citizens might also be able to provide CA services.** Consider, for example, large health insurance providers or health maintenance organizations. Such organizations routinely establish basic identity information on their members and patients. Increasingly, these organizations may desire to communicate sensitive information (e.g. diagnostic test results, payment information, and appointment verifications) to doctors and patients electronically, and they may develop secure communications systems for their own purposes. Electronic identities established for these purposes might be sufficiently reliable for the transmission of sensitive government information.

In carrying out their missions, some government agencies and quasi-governmental entities have frequent or regular interactions with large numbers of citizens. They may, therefore, be plausible candidates for providing CA services to a broad population.

## 7. GETTING FROM THE CURRENT MAILING SYSTEM TO THE OFFICIAL E-MAIL EXCHANGE SYSTEM

Secure e-mail communication between government agencies and individual citizens will not become a reality overnight. Considerable groundwork must be laid: standards for privacy, integrity, and authentication must be established; certificate authorities must be identified or established; a host of institutional, administrative, and policy questions have to be resolved; and, most important, accumulating experience and maturing laws, regulations, and practice norms will have to provide a foundation for trust in using official e-mail exchange system for sensitive communications. The task of creating a capability for secure communication between governments and citizens can be accomplished by [7]:

- **An incremental, experimental approach.** The likelihood that we will get the system entirely right on the first try is vanishingly small, and there is little point in trying at the outset for a system that will meet all government demands. Much better to concentrate on functional requirements, and to experiment, starting with relatively undemanding applications and relatively no sensitive information, and then to gradually strengthen systems and procedures until we are confident that we can handle the most complex transactions and the most sensitive data.

- **Citizens should be able to "opt in."** At least during a transition period when the security and reliability of on- line communication with government agencies is still being demonstrated, citizens must be able to "opt in" to such communications arrangements, positively choosing for their records or accounts to be accessible on-line. It is unreliable to assume that citizens have sufficient understanding of the implications of on-line access and of procedures to control this access to make on-line access the default option.

- **"Out of band" communication will continue to be important.** To provide adequate assurance of the identity of an individual, it is often useful to use a separate channel of communication for verification. For example, although application for a digital identity certificate might be made on-line or in person, the password or personal identification number (PIN) unlocking or activating the certificate might be sent by postal mail to the correspondent's registered home address. All of this suggests that policy should aim to maintain and to utilize multiple channels for electronic communication: the Internet, automated telephone services, bank ATM networks, and the like.

- **Success will depend on education and training**. Successful development and deployment of mechanisms for digital communications between citizens and governments will require extensive efforts to educate citizens regarding the advantages of new communications modes and associated protections for sensitive information. Training in how to establish, use, and

protect a digital identity will also be key. Equally important will be establishing realistic expectations among users; just because e-mail can be transmitted nearly instantaneously.

## 8. CONCLUSION

This paper considered the problem of using e-mail as an official communication tool. This problem should be investigated from different prospective. Here, only technical one has been considered. This study presented that the adoption of e-mail as an official communication is legally tolerable. Also, this study has suggested a reliable e-mail system that can be applied as an official communication, which has met all the requirements, by deploying a number of security mechanisms.

## 9. REFERENCES


[1] Connell S., and Galbraith I. A."*The Electronic Mail Handbook: A Revolution in Business Communications*", Great Britain, Kogan Page Ltd, (1982)

[2] Hayden M., and Brad H. "*The On-Line/E-Mail Dictionary*", New York, Berkley Publishing Group, (1997)

[3] Abbate J. "*Inventing the Internet*", Boston, MIT Press, (1999)

[4] Lerner M. How Stuffs Work [Online]. Available at: http://www.learnthenet.com/english/html/20how_2.htm, [Accessed April 2008]

[5] Whelan J. "*E-mail @ Work*", Great Britain, Biddles Ltd, (2000)

[6] Computer Desktop Encyclopedia. Electronic mail [Online]. Available at: http://www.answers.com/Electronic+mail?cat=technology [Accessed May 2008]

[7] Neu C. R., Anderson R. H., and Bikson T. K. E-Mail Communication between Government and Citizens - Security, Policy Issues, and Next Step. RAND science and technology organization [Online]. Available at: http://www.rand.org/pubs/issue_papers/IP178/index2.html, [Accessed April 2008]

[8] U.S. Department of Labor. Occupational Safety & Health Administration Public Website. OSHA E-Correspondence system [Online]. Available at: http://www.osha.gov/, [Accessed April 2008]

[9] Perry T. S. "*Forces For Social Change".* IEEE Spectrum, 29(10):30 – 32, 1992

[10] Yayehyirad Kitaw. "*E-Government in @frica Prospects, challenges and practices*". Swiss Federal Institute of Technology in Lausanne (EPFL), 2006

[11] Abu Al Lail E. A. "*Legal issues of eTransactions*". Kuwait, Scientific Research Council, 2003

[12] Dam K.W. and Lin H.S. "*Cryptography's Role in Securing the Information Society*". National Research Council, Washington D.C., 1996

[13] Schneier B. "*Applied cryptography*", 2ed edition, New Jersey, John Wiley and Sons, (1996)

[14] Forouzan B. A. "*Cryptography and network security*", New York, McGraw-Hill, (2008)

[15] Stallings W. "*Cryptography and Network Security*", 4th edition, New Jersey, Prentice-Hall, (2006)

[16] Budd C. "*Exchange Server 2003 Message Security Guide*", Microsoft, (2004)

[17] Charles E. M. "*Ensuring Electronic Mail System Delivery Capability*". In Proceedings of the IEEE military communications conference. Atlantic City NJ, 1999

[18] Mailinfo Ltd. Don't let your emails get lost in spam! [Online]. Available at: http://www.mailinfo.com/web [Accessed May 2008]

# A Lower Bound Study on Software Development Effort

**Lung-Lung Liu** llliu@mail.mcu.edu.tw
*International College Ming Chuan University*
*Gui-Shan, Taoyuan County,*
*Taiwan, ROC 333*

**Abstract**

This paper depicts a study on the lower bound of software development effort. The work begins with the transformation model which is with the popular software development lifecycle. In general, a combination of properly handled transformations can ultimately produce the software, and the transformations form the path of software development. Each transformation is associated with its effort, or the weight, in a path. There can be a number of these paths since many different methods and tools can be used or reused for a specific software development case. Then, the Shortest Path algorithm is applied to find a shortest path which is with a minimal total effort among all the paths. However, from time to time, when advanced methods and tools are introduced, the new paths and efforts will change the previously identified shortest path. Hence, the continued work is to discuss the minimal total effort of a potential shortest path, although it may be currently unavailable. Convergence analysis is firstly provided for the discussion of whether this shortest path exists, and lower bound analysis is then provided for the discussion of completeness and soundness. This lower bound study is significant, since an optimal software development effort is determinable.

**Keywords:** Lower Bound, Software Development, Effort

## 1. INTRODUCTION

Software development effort has been a research topic since the early 1980's [1]. Most of the related studies are on the estimation of the effort, or the cost, of software development. In the review paper [2], it is indicated that there have been great contributions in this field. Some of them came from the academic, and some others came from the industry. In the other paper [3], a debate on "Formal Models or Expert Judgment?" is in discussion, and this is a typical example that to both researchers and practitioners, the effort issue is interesting. The estimated result is usually a predicted range of values (such as 200~250 person-months), and they can be obtained by applying probabilistic models and referencing historical data. For example, if a team has been working together in similar projects for five years, then their coming year's productivity can be easily predicted by applying a simple curve fitting approach. However, as team members are to change, project styles are to change, and working environments are to change, there are more and more factors to take into consideration. The probabilistic models assume that the factors are random variables with proper distribution functions and then perform the calculation. Although these models are just at the entry level to the studies, these seemed-to-be-straight descriptions may have blocked the software engineering practitioners to pay attention to the estimation fundamentals. They are complicate, and they are hard to be practiced.

In this paper, we are to depict a lower bound study on software development effort. It is with a computer science oriented algorithmic approach [4] but not the traditional probabilistic model approach. We begin with a transformation model which is based on the popular software development lifecycle and the programming language support systems. Successful and efficient transformations are mechanisms such as the compilers. They are able to change the external forms of software, while the internal computation essentials are still the same. Hence, when several transformations are performed, an executable image which is equivalent to an originally specified requirement is expected, and it can be loaded into a computer for a run. Every transformation is with an effort. If it is close to zero, then the transformation is efficient. On the other hand, if it is a relatively large value, then perhaps the project is to fail and alternatives should be considered. According to the transformations applied to a specific development case, a path through the transformations can be drawn. We also put the effort of each transformation as the value associated to the element in the path. Since there are many possible development cases, there are many paths. Now it is the Shortest Path problem, and the well-known Dijkstra's algorithm [5] can be used to find the shortest path. If we can prove that the shortest path does exist among these possible development cases, then there is the lower bound.

The proof is with the discussion on whether the approaches applied in a transformation is convergent. For example, iterations are frequently introduced in software development methods. However, without the proper control, the iterative processes may cause an infinite effort. A path with this transformation will never be the shortest path. On the other hand, since nowadays advanced software tools for efficient development methods are publically available, there are transformations that their efforts are close to zero. We want to identify these transformations and set them aside, and we are to check the remaining for whether further analysis is still doable. Theoretically, with a full set of reusable modules (which can be quite large) ready, the lower bound of software development effort can be with linear complexity, counted on the number of possible requirements specified. In the following, the transformation model is introduced in Section 2, and the applying of the Shortest Path algorithm is described in Section 3. The convergence analysis and lower bound analysis are provided in Sections 4 and 5, and the requirement specification that dominates the bound is discussed in Section 6. Finally, the conclusions are in Section 7.

## 2. THE TRANFORMATION MODEL

The set of integrated compile-link-load processes is a typical example for the transformation model. Let abc.c be a source program written in C language. With a C compiler for a specific development environment, abc.c can be transformed (compiled) into abc.obj, or an object program. With a link editor (or other equivalent tools), abc.obj can be transformed (added with other necessary object programs externally referenced) into abc.exe, which is a ready to run loadable module. Finally, with a loader, the load module is transformed (for relative address resolution, basically) into an image of a memory map and then copied into the memory for a real run. The above mentioned source program, object program, load module, and image are totally different in their formats, but actually they are the same from an essential software point of view. They are with exactly the same logical sequences of machine instructions that are for a desired computation goal. If they are not, then the language subsystem supported for that development environment is incomplete. The transformations are depicted in Figure 1.

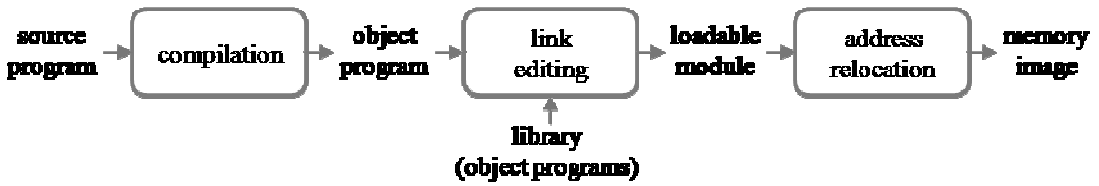



**FIGURE 1:** Program transformation

In the popular software development lifecycle [6], there are stages. The stages can be individually suggested, depending on the approaches or methodologies applied. For example, in a common Waterfall based approach, they may be the requirement analysis stage, the design stage, the implementation stage, and the maintenance stage. Continuing with the previous compile-link-load case, we only take the first three of the stages into consideration, since the last one is currently out of our scope. The associated transformations are to ultimately produce the source program abc.c (after that the language subsystem will properly handle the following details). In their natural sequences, the major functions of the transformations are to (1) transform the actual requirement in stakeholders' mind (that is in the cloud) into a requirement specification (that must be in machine readable format for further automation), (2) transform the requirement specification into a design specification, and (3) transform the design specification into a pseudo code, or equivalently, the source program. The transformations are depicted in Figure 2.


**FIGURE 2:** Specification transformation

We check the transformations one by one, starting from the last (which is the transformed source program) to the first, to see the difficulties. In transformation (3), current methods and tools can easily help the works, since there are polynomial time algorithms to solve tree traversal problems and grammar related parsing problems. The assumptions here are that the design specification has clearly indicated the number of modules, their interfaces to one another, and their computation basics with data structures, and that the programming language used is with strict syntax rules. The cases in transformation (2) are complicated. A key problem is that the common requirements specified are with a network structure, since they can be related to one another or they are independent. If the modules in the design specification are expected to be with a hierarchical structure (such that the programs can be with a perfect top-down tree structure), then we need an algorithm to convert a thing in network structure into some other thing in an equivalent tree structure. However, there is no such an algorithm. Software developers are used to taking a heuristic approach (with their experience) to try to find proper groups of requirements and let them be in a hierarchy. In some of the (worst) cases, the proper groups could never be found. On the other hand, if the modules in the design specification are not expected to be in any structure, or they are still in a network structure, then the works are easy but a non-procedural programming language support is necessary in the following development stags. In transformation (3), the actual work is requirement elicitation, which is to put outside information (in the stakeholders' mind, actually) into the computer as the first version of digitized (machine readable) data for further software development details. Before that, the computers can do nothing since there is no data to process. The transformation model is a basis for the study of software development effort, since ultimately programs are produced with a sequence of proper transformations performed.

## 3. APPLYING THE SHORTEST PATH ALGORITHM

Transformations are with efforts, and a sequence of transformations is with a total effort. Let $e_i$ be the effort of performing transformation *i*. If in a graph the nodes represent the things before and after a transformation, and the edges (with direction) represent the effort of the transformation, then a sequence of transformation can be a *path* made of efforts associated. An example of the path made of transformations mentioned earlier is given in Figure 3.

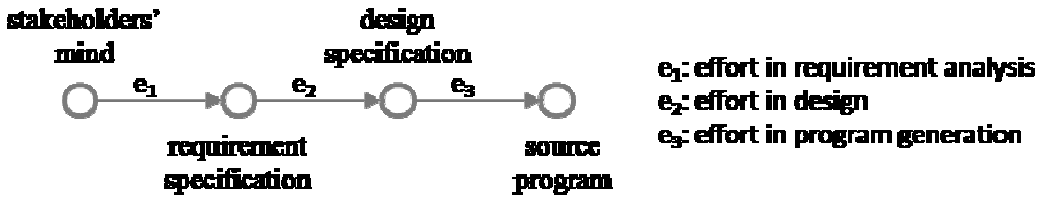



**FIGURE 3:** A path made of transformations

Occasionally, there are multiple paths, since there are different methods and tools that can be applied to a specific transformation. The individual efforts are different, and the total efforts are different. In addition, a transformation may be skipped when an advanced methodology is applied (such as an interpreter is used for a compiler in the language support subsystem). At this moment, we temporarily focus on the topic that multiple paths are there, and that why they are available is to be discussed in the next section. An example of multiple paths with a same sequence of transformations (same methodology used, but different methods used in a transformation) is provided in Figure 4, and another example of multiple paths with different sequences of transformations (different methodologies used, hence a transformation may be skipped) is provided in Figure 5. In the first example, there are three paths, with efforts of $e_{11}+e_{12}+e_{13}$, $e_{21}+e_{22}+e_{23}$, $e_{31}+e_{32}+e_{33}$, respectively. In the second example, there are four paths, with efforts of $e_1+e_2+e_3$, $e_1+e_5$, $e_4+e_3$, $e_6$, respectively. There may be more complicated multiple paths, since a combination of the cases provided in the example are all possible.

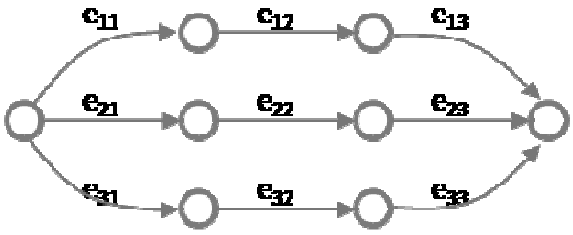



**FIGURE 4:** Multiple paths with a same sequence of transformations

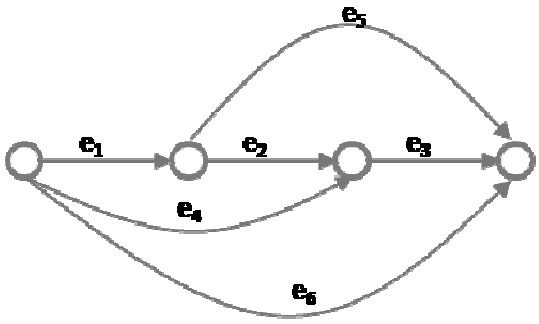



**FIGURE 5:** Multiple paths with different sequences of transformations

When paths are well defined, we are to find a path which is with the least effort among all of them. It is natural that Dijkstra's famous Shortest Path algorithm can be applied for a solution. The problem can be solved easily, but there are conditions. For example, if in every path there is a different effort which is with infinite as its value, then every path is with a total effort of infinite and there will be no shortest path at all.

## 4. CONVERGENCE ANALYSIS

The major purpose of doing convergence analysis is to make sure that the Shortest Path algorithm work well and hence is useful. There are at least two factors as conditions that should be taken into consideration: the possible values of every individual effort *e*, and the way the computation like "$e_1+e_2$" is to be performed. If both of the factors are with the convergence features, then the shortest path exists.

First, the value of an effort should be greater than zero. An effort with a negative value means that a transformation can even recover the resources (such as man-power, time, and money) exhausted before, but it is physically impossible since times elapse only. A zero value effort is a redundant transformation which should be removed from the associated path. However, an effort with a positive value close to zero may mean that the transformation has fully utilized the skills of tool automation, and it is encouraged. On the other hand, an effort with a relatively large may mean that the transformation is not efficient, and it is to be improved. The worst case is that the value is close to, say, infinite. Although it is only mathematically possible (infinite does not exist practically), it happened and project failed due to poor transformation (or no convergence management) works were performed.

The poor cases are popularly available, and from the convergence point of view, they really offend. There are typical examples, and we propose three of them. The iteration based methods are risky, since works may be performed repeatedly but with no progress. A criterion to stop the iteration should be set up in advance (even it is budget oriented), just like the defining of termination conditions in a recursive procedure. The introducing of the divide-and-conquer strategy may be another trap in some of the poor cases. The divide part makes the developer a significant progress, but, after the conquer part there always is a merge part which is rarely emphasized. When the effort of performing a merge is greater than that of solving the original problem, the convergence is broken. Besides, only independent works are suggested to be divided and conquered, otherwise the effort on synchronization is added in. The last one is related to the debugging work, and that happened to the junior programmers. Once an error in a program was found and is to be debugged, there is the chance that the error was corrected while new errors are added. The total number of errors is increasing but not decreasing, hence the transformation is not with convergence.

Secondly, the way the computation like "$e_1+e_2$" should be performed simply. For example, it is just a normal addition of numbers. There is no complicated definition on the operator "+" and there is no specific domain for the operands. To control the conditions related to this factor, the immediate consideration is that the transformations must be independent hence the computation of the joint efforts of two consecutive transformations can be straight. Otherwise, we have to clarify the dependency features and then see whether they can be calculated. The standards of CASE tools interconnections [7] are the best references for this. The standards define the common data formats between two consecutive transformations and the disciplines to access the data. Following the standards to apply the methods and tools in the transformations will guarantee a smooth and costless interconnection. This also means that there is no extra effort on data conversion, and the "+" operation is the simplest arithmetical addition.

## 5. LOWER BOUND ANALYSIS

The purpose of providing the lower bound analysis is to formally prove the logical completeness and soundness in this study. In the following paragraphs, Facts and Theorems are given. The Facts are always true, and they are with explanations only. Theorems are with proofs.

**Fact 1.** The transformation model works.

That is, the belief of software development lifecycle is feasible by applying the model. Let $T$ be the set of transformations, and let $t$ in $T$ be a transformation that takes any $d$ in domain $D$ and then transforms $d$ into a specific $r$ in range $R$. In short, $t : D \rightarrow R$, and $r = t(d)$. If both $t_i$ and $t_j$ are in $T$, then the result of consecutive transformations $t_i$ and $t_j$ performed to $d_i$ is $t_j(t_i(d_i))$. A sequence of transformations, such as the requirement analysis, design, and implementation stages suggested in the conventional software development approaches, really ultimately generate the source programs. We leave the definitions of the sets $T$, $D$, $R$ free.

**Fact 2.** A path representing the sequence and efforts of transformations performed in a specific software development case is determinable.

That is, a path is also with the transformation model but in a different format used in graphs. Let $r = t(d)$ be with the definitions used previously. Equivalently, $t$ can be represented by using an edge $e$ connecting nodes $d$ and $r$, with a direction, and the value associated with $e$ is the effort of performing transformation $t$. Then, a sequence of $n$ transformations $t_1$, $t_2$, $t_3$, …, $t_n$ firstly performed to $d$ is a path $p : e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow \ldots \rightarrow e_n$, started at $d$ and ended at the lastly generated (what was transformed into) specific $r$. The total effort of the path $p$ can be derived from $e_1$, $e_2$, $e_3$, …, $e_n$.

**Fact 3.** The shortest path from node $a$ to node $b$ represents the least effort among all possible software development cases targeted for a specific application, given $a$ as the initial input and $b$ as the final output.

That is, the least effort can be determined by finding the shortest path among all the possible paths derived from those cases. In other words, within two different paths, if they have common joint nodes (different transformations were performed for a common output, and then other different transformations continued), then there can be a combined new path which is with the first half sequence of one path and the second half sequence of the other. However, the new path is with less effort than the previous two. Let $p_1 : e_1 \rightarrow e_2$ and $p_2 : e_3 \rightarrow e_4$ be the two paths. The case happens when there is a joint node at the middle of these two paths, with the condition that $e_1 + e_4 < e_1 + e_2$ and $e_1 + e_4 < e_3 + e_4$.

**Theorem 1.** The Shortest Path algorithm can be applied to find the least effort of software development cases to a specific application.

Proof: According to Fact 1, the transformation model is applicable to software development processes, starting from the requirement specification stage for the source programs. According to Fact 2, the path representing a sequence of transformations can be properly defined in a graph, with transformation efforts as weights of the edges. According to Fact 3, the shortest path is the sequence of transformations with the least software development effort among all the possible cases. Let $G = \{V, E\}$, where $V$ is the set of nodes and $E$ is the set of edges with weights representing the transformation efforts, be the graph derived from the software development cases to a specific application. Since the Shortest Path algorithm is bounded by the number of nodes $V$ in the graph (the computation time is of order $V^2$ or $V \log V$), the finding of the shortest path is guaranteed. The total weight, cumulated by those associated to the edges forming the path, is the least effort of the software development cases.

**Theorem 2.** The lower bound of software development effort can be found if all the development cases are collected.

Proof: Let there be $n$ possible paths according to all the cases. By Theorem 1, the Shortest Path algorithm can be applied to find the least effort $e_s$. That is, $e_s < e_i$, i=1, $n$, i ≠ s. The lower bound L is $e_s$, since for every $e_i$, i=1, $n$, L ≤ $e_i$.

**Fact 4.** The effort of using tools is close to zero.

That is, when tools are used, there is almost no effort, compared with the barehanded way to do the development work. Here the effort is only on the using but not the introducing of tools. Tools are software running on computers, and the performance is faster than human in millions of times. We can conclude that the effort of using tools is close to zero.

**Theorem 3.** By fully applying the tools, the lower bound of software development effort is with a linear complexity, counted on the number of possible requirement specified.

Proof: Let there be powerful tools and software development environment supports, although some of them are currently unavailable. Let the shortest path found be made of only one edge, and it is associated with the least effort among all edges. This has actually indicated an efficient and effective methodology that is able to directly transform the requirement specified into a source program. Since the effort of using tools is close to zero according to Fact 4, we neglect them. Then the effort is to let the tools be able to recognize the possible requirements. Let the requirements be specified in a yes/no feature list, and let the number of the features in the list be *n*. The effort of handling all the possible requirements specified is with a linear complexity of *n*.

## 6. REQUIREMENT SPECIFICATION IN SELECTION

Works related to requirement analysis are actually with the most effort in software development [8], since the works at other development stages have almost been fully automated by using tools or even skipped. For example, there are very high level requirement specification languages and the support systems that can accept requirements as statements in a special syntax, such as the Prolog programming language and her runtime environment support. In Prolog, a requirement can be represented by a number of logical rules, and an equivalent number of statements directly translated from the rules actually completed the source program, or the software development works. The other example is the object oriented development methods. Usually, at the end of object oriented analysis stage, use cases should have been generated, and then at the design stage, all the necessary objects are to be determined. If the class libraries are well classified and indexed, then that will be just a mapping work. (However, the classification problems or the domain analysis issues are beyond the discussion scope here.) These two examples have indicated that the effort for transformation from a dataflow diagram to a structured chart in the conventional design stage could be totally skipped.

The focus has been on the minimization of the scope that developers should really pay attention to, and it is in the handling of requirement analysis. There are cases that, for some specific requirements, what actually in the stakeholders' minds are unclear, even they themselves cannot tell. Conventionally, there are at least three necessary steps in the requirement analysis stage. The first one is requirement elicitation, the second one is requirement integration, and the last one is requirement negotiation. It is a discipline that requirements ought to be specified as items (such as use cases), and they are countable by numbers. Then, the three steps can be more precise. Requirement elicitation means a result of identifiable items shown in a list or a network structure. Requirement integration means a result of being integrity among the items. Requirement negotiation means a result of practical (developable) items after conflict resolution is done. The key is, the numbers of items mentioned above are all countable.

If there is another discipline says that all the requirement items specified should be flat, or with no hierarchy among them, then potentially the tools can do more. Let there be a rich reusable library of software components, supported by an intelligent configuration mechanism. A tool is suggested to ask *n* yes/no questions about a targeted application, and answers are expected. The number of the combination of all the possible answers is $2^n$. If the tool is able to provide all of the source programs according to all of the possible answer sets, then the software development is done. Although the number $2^n$ is of exponential, practically when *n* can be controlled in an affordable range, the providing of such tools is considerable. In this way, the only effort in requirement

analysis stage is to do requirements specification in selection. It is to make decisions on the selecting of yes/no answers to the *n* questions. The complexity of the effort is of linear to *n*.

## 7. CONCLUSIONS

We briefly (1) specified the transformation model in software development, (2) mapped the transformations into a graph where a path representing the total efforts of a sequence of transformations, and (3) suggested that the Shortest Path algorithm can be applied to find the shortest path (if it exists) which represents the least effort among all the development cases to a specific application. We also provided the convergence analysis to discuss the possible infinite efforts due to iterative but not effective processes performed in the transformations, because if the convergence management is not considered, then the shortest path may not exist. Finally, we provided the lower bound analysis to show the logical completeness and soundness of the whole study. We are to conclude that once all of the possible development cases (including the currently unavailable ones) are checked, the shortest path found among the mapped paths is the one with least effort, and this least effort is the lower bound.

Current results indicated that the lower bound can be with a linear complexity, counted on the number of requirements specified. The assumption is that the reusable modules and the configuration mechanisms are powerful enough, just like the customers are to configure an operating system on their computers. They can obtain the custom-made software by simply setting the parameters provided in a customizable profile. The configuration tools will provide a combination of modules that fits the requirements of the desired operating system. The original software development problem is with a complicated complexity, but the tools may have solved most of them. For example, they absorb the ones with exponential complexity, while leave the one with linear complexity to the developers. However, these are the currently unavailable ones.

The future works include (1) the analysis of the conditions that may hold a minimum effort such that the lower bound is thus blocked, (2) the analysis of the cases that methods in transformations did not really reduce the effort but just postpone the works, and (3) the discussion for possible further formal proofs.

## 8. REFERENCES

[1] R. W. Selby and B. W. Boehm. *"Software Engineering: Barry W. Boehm's Lifetime Contributions to Software Development, Management, and Research".* Wiley-IEEE, 2007

[2] M. Jørgensen, M. Sheppered. *"A Systematic Review of Software Development Cost Estimation Studies".* IEEE Transactions on Software Engineering, 33(1):33-53, 2007

[3] M. Jørgensen, B. Boehm, S. Rifkin. *"Software Development Effort Estimation: Formal Models or Expert Judgment?"* IEEE Software, 26(2):14-19, 2009

[4] D. E. Knuth. *"The Art of Computer Programming, Volume 4, Fascicle 0: Introduction to Combinatorial Algorithms and Boolean Functions".* Addison-Wesley Professional, 2008

[5] Dijkstra, E. W. *"A Note on Two Problems in Connexion with Graphs, in Numerische Mathematik, Vol. 1".* Mathematisch Centrum, Amsterdam, the Netherlands, pp. 269-271, 1959

[6] ISO/IEC 12207. *"Information Technology – Software Lifecycle Processes".* IEEE/IEC Standard 12207.

[7] *"IEEE Guide for CASE Tool Interconnections – Classification and Description (IEEE std 1175)".* IEEE Standards, Available at: http://ieeexplore.ieee.org/.

[8] J. Lee and N. L. Xue. *"Analyzing User Requirements by Use Cases: A Goal-Driven Approach"*. IEEE Software, 1999

# Steganography Using Dictionary Sort on Vector Quantized Codebook

**Dr. H. B. Kekre** hbkekre@yahoo.com
*MPSTME, NMIMS University, Vile-parle (W), Mumbai-56, India*

**Archana Athawale** athawalearchana@gmail.com
*Ph.D. Scholar MPSTME, NMIMS University, Mumbai-56, India Assistant Professor, Thadomal Shahani Engineering College, Bandra (W), Mumbai-50, India*

**Tanuja Sarode** tanuja_0123@yahoo.com
*Ph.D. Scholar MPSTME, NMIMS University, Mumbai-56, India Assistant Professor, Thadomal Shahani Engineering College, Bandra (W), Mumbai-50, India*

**Sudeep Thepade** sudeepthepade@gmail.com
*Ph.D. Scholar MPSTME, NMIMS University, Mumbai-56, India Assistant Professor, MPSTME, NMIMS University, Mumbai-56, India*

**Kalpana Sagvekar** kalpanasagvekar@gmail.com
*Lecturer, Fr. Conceicao Rodrigues College of Engineering, Bandra (W), Mumbai-50, India*

**Abstract**

This paper presents a new reversible data hiding scheme using vector quantization (VQ). In traditional VQ based data hiding schemes secret data is hidden inside index based cover image resulting in limited embedding capacity. To improve the embedding capacity as well as to have minimum distortion to carrier media our method proves good. In this paper we have used four different codebook(CB) generation algorithms Linde Buzo and Gray (LBG), Kekre's Proportionate Error (KPE), Kekre's Median Codebook Generation algorithm (KMCG) and Kekre's Fast Codebook Generation Algorithm (KFCG) to prepare codebooks. So the Herculean task of increasing data hiding capacity with minimum distortion in recovered secret message is achieved with help of proposed techniques which are more robust against stegaanalysis.

**Keywords:** Data Hiding, VQ, LBG, KPE, KMCG, KFCG.

# 1. INTRODUCTION

The Internet has revolutionized the modern world and the numerous Internet based applications that get introduced these days add to the high levels of comfort and connectivity in every aspects of human life. As of September 2009, approximately 1.73 billion people worldwide use Internet for various purposes – ranging from accessing information for educational needs to financial transactions, procurement of goods and services [1]. As the modern world is gradually becoming "paperless' with huge amount of information stored and exchanged over the Internet, it is imperative to have robust security measurements to safeguard the privacy and security of the underlying data.

Cryptography techniques [2] have been widely used to encrypt the plaintext data, transfer the ciphertext over the Internet and decrypt the ciphertext to extract the plaintext at the receiver side. However, with the ciphertext not really making much sense when interpreted as it is, a hacker or an intruder can easily perceive that the information being sent on the channel has been encrypted and is not the plaintext. This can naturally raise the curiosity level of a malicious hacker or intruder to conduct cryptanalysis attacks on the ciphertext (i.e., analyze the ciphertext vis-à-vis the encryption algorithms and decrypt the ciphertext completely or partially) [2]. It would be rather more prudent if we can send the secret information, either in plaintext or ciphertext, by cleverly embedding it as part of a cover media (for example, an image, audio or video carrier file) in such a way that the hidden information cannot be easily perceived to exist for the unintended recipients of the cover media. This idea forms the basis for Steganography, which is the science of hiding information by embedding the hidden (secret) message within other, seemingly harmless images, audio, video files or any other media. Steganography protects the intellectual property rights and enables information transfer in a covert manner such that it does not draw the attention of the unintended recipients.

The existing schemes of data hiding can roughly be classified into the following three categories:
Spatial domain data hiding [3, 4, 5]: Data hiding of this type directly adjust image pixels in the spatial domain for data embedding. This technique is simple to implement, offering a relatively high hiding capacity. The quality of the stego image can be easily controlled. Therefore, data hiding of this type has become a well known method for image steganography.

Frequency domain data hiding [6, 7]: In this method images are first transformed into frequency domain, and then data is embedded by modifying the transformed coefficients.

Compressed domain data hiding [8, 9]: Data hiding is obtained by modifying the coefficients of the compressed code of a cover image. Since most images transmitted over Internet are in compressed format, embedding secret data into the compressed domain would provoke little suspicion.

In steganography the thirst is on increasing secret data hiding capacity and making the steganography techniques more and more robust against stegaanlysis attacks.

# 2. VQ COMPRESSION TECHNIQUE

Vector Quantization (VQ) [9-14] is an efficient technique for data compression [31-34] and is very popular in a variety of research fields such as data hiding techniques [7,8], image segmentation [23-26], speech data compression [27], content based image retrieval CBIR [28, 29] and face recognition [30].

## 2.1 Codebook Generation Algorithms

### 2.1.1. Linde-Buzo-Gray (LBG) Algorithm [10]

In this algorithm centroid is calculated as the first codevector for the training set. In Figure 1 two vectors v1 & v2 are generated by using constant error addition to the codevector. Euclidean

distances of all the training vectors are computed with vectors v1 & v2 and two clusters are formed based on nearest of v1 or v2. This procedure is repeated for every cluster. The drawback of this algorithm is that the cluster elongation is $-45^{o}$ to horizontal axis in two dimensional cases. Resulting in inefficient clustering.

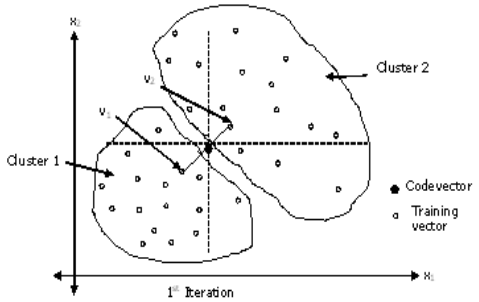



**FIGURE 1:** LBG for 2 dimensional case

### 2.1.2. Proportionate Error Algorithm (KPE) [11], [12]

Here proportionate error is added to the centroid to generate two vectors v1 & v2. Magnitude of elements of the centroid decides the error ratio. Hereafter the procedure is same as that of LBG. While adding proportionate error a safe guard is also introduced so that neither v1 nor v2 go beyond the training vector space. This removes the disadvantage of the LBG. Both LBG and KPE requires 2M number of Euclidean distance computations and 2M number of comparisons where M is the total number of training vectors in every iteration to generate clusters.

### 2.1.3. Kekre's Median Codebook Generation Algorithm (KMCG) [13]

In this algorithm image is divided into blocks and blocks are converted to the vectors of size k. The Figure 2 below represents matrix T of size M x k consisting of M number of image training vectors of dimension k. Each row of the matrix is the image training vector of dimension k.

$$T = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,k} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,k} \\ . & . & . & . \\ . & . & . & . \\ x_{M,1} & x_{M,2} & \cdots & x_{M,k} \end{bmatrix}$$

**FIGURE 2:** Training Matrix

The training vectors are sorted with respect to the first member of all the vectors i.e. with respect to the first column of the matrix T and the entire matrix is considered as one single cluster. The median of the matrix T is chosen (codevector) and is put into the codebook, and the size of the codebook is set to one. The matrix is then divided into two equal parts and the each of the part is then again sorted with respect to the second member of all the training vectors i.e. with respect to the second column of the matrix T and we obtain two clusters both consisting of equal number of training vectors. The median of both the parts is the picked up and written to the codebook, now the size of the codebook is increased to two consisting of two codevectors and again each part is further divided to half. Each of the above four parts obtained are sorted with respect to the third column of the matrix T and four clusters are obtained and accordingly four codevectors are obtained. The above process is repeated till we obtain the codebook of desired size. Here quick sort algorithm is used and from the results it is observed that this algorithm takes least time to generate codebook, since Euclidean distance computation is not required.

### 2.1.4. Kekre's Fast Codebook Generation (KFCG) Algorithm[14]

In [14], KFCG algorithm for image data compression is proposed. This algorithm reduces the time for codebook generation. It does not use Euclidian distance for codebook generation. In this algorithm image is divided into blocks and blocks are converted to the vectors of size k. Initially we have one cluster with the entire training vectors and the codevector $C_1$ which is centroid.
In the first iteration of the algorithm, the clusters are formed by comparing first element of training vector with first element of codevector $C_1$. The vector Xi is grouped into the cluster 1 if $x_{i1} < c_{11}$ otherwise vector $X_i$ is grouped into cluster 2 as shown in Figure 3(a). where codevector dimension space is 2.

In second iteration, the cluster 1 is split into two by comparing second element $x_{i2}$ of vector Xi belonging to cluster 1 with that of the second element of the codevector which is centroid of cluster 1. Cluster 2 is split into two by comparing the second element $x_{i2}$ of vector $X_i$ belonging to cluster 2 with that of the second element of the codevector which is centroid of cluster 2, as shown in Figure 3(b).

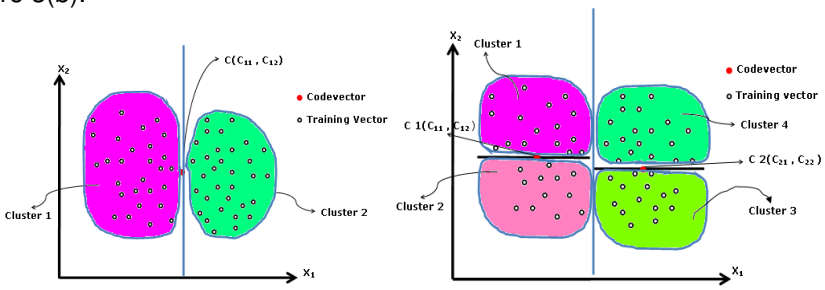



**FIGURE 3:** KFCG algorithm for 2-D case

This procedure is repeated till the codebook size is reached to the size specified by user. It is observed that this algorithm gives less error as compared to LBG and requires least time to generate codebook as compared to other algorithms, as it does not require computation of Euclidian distance.

## 3. PROPOSED APPROACH

In this approach, we are hiding the secret data into codebook generated using various codebook generation algorithm such as LBG[10], KPE[12][13], KMCG[14], KFCG[15]. There are various ways of hiding: 1bit, 2 bits, 3 bits, 4 bits & variable bits hiding.

The algorithm is as follows:

a. Divide the image into 2×2 block of pixels.
b. Generate initial cluster of training set using the rows of 12 values per pixel window.
c. Apply codebook generation algorithm LBG/KPE/KFCG/KMCG on initial cluster to obtain codebook of size 2048 codevectors.
d. Add initial index position column in codebook (CB).
e. Perform dictionary sort on CB.
f. Hide data into sorted CB except the last column.
g. Again add final index position column in Stego CB.
h. Sort the stego CB.
i. From sorted CB reconstruct the image.
j. Send the reconstructed image & new final index position to receiver.
k. Receiver will divide image into blocks generating training vectors. Collection of unique training vector is nothing but CB.
l. Arrange the entries of codebook using final index position
m. Extract the secret data.

### 3.1 Variable Bit Hiding Algorithm

For variable bit hiding Kekre's algorithm [3] is used.

a. If the value of codebook vector element is in the range 240 ≤ gi ≤ 255 then we embed 4 bits of secret data into the 4 LSB's codebook vector element. This can be done by observing the 4 most significant bits (MSB's). If they are all 1's then the remaining 4 LSB's can be used for embedding data.
b. If the value of codebook vector element is in the range 224 ≤ gi ≤ 239 then we embed 3 bits of secret data. . This can be done by observing the 3 most significant bits (MSB's). If they are all 1's then the remaining 3 LSB's can be used for embedding data.
c. If the value of codebook vector element is in the range 192 ≤ gi ≤ 223 then we embed 2 bits of secret data. . This can be done by observing the 2 most significant bits (MSB's). If they are all 1's then the remaining 2 LSB's can be used for embedding data.
d. If the value of codebook vector element is in the range 0 ≤ gi ≤ 191 we embed 1 bit of secret data.

## 4. RESULTS AND EVALUATION

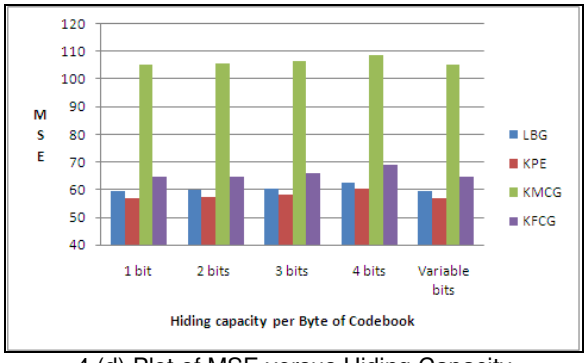
In our proposed approach, we have generated codebook using LBG, KPE, KMCG and KFCG for 24 bit color image of size 256×256 shown in Figure 4(a) to 8(a). Codebook is of size 2048×12 (i.e. 2048 code vectors each contains 12 bytes - 4 pairs of RGB). We have hidden 32×32 gray image.

Figure 4 to Figure 8 Shows the results of 1 bit, 2 bits, 3 bits, 4 bits and Variable bits using codebook obtained from LBG, KPE, KMCG and KFCG on the various cover images Eleafrica, Bird, Panda, Flowers and Manydogs hiding same secrete image for fair comparison respectively.

4 (a) Original Cover image Eleafrica.bmp


4 (b) Secret Message

4 (c) Reconstructed image using KPE for Variable bits Method



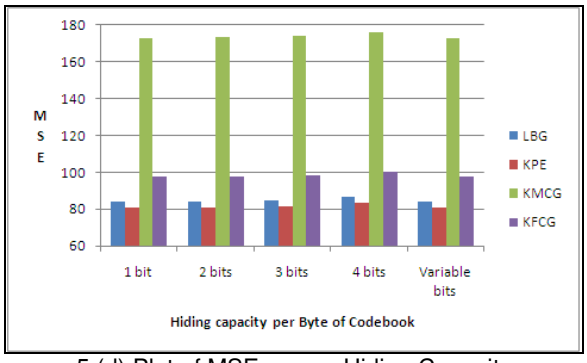

4 (d) Plot of MSE versus Hiding Capacity

**FIGURE 4:** Results of 1bit, 2bits, 3 bits, 4bits and Variable bits on the cover image Eleafrica and secrete image shown in Figure 4(b).

5 (a) Original Cover image Bird.bmp

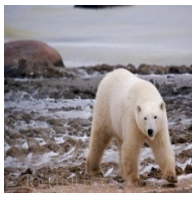
5 (b) Secret Message

5(c) Reconstructed image using KPE for Variable bits Method



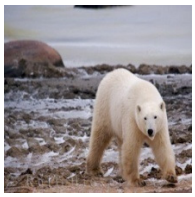

5 (d) Plot of MSE versus Hiding Capacity

**FIGURE 5:** Results of 1bit, 2bits, 3bits, 4bits and Variable bits on the cover image Bird and secrete image shown in Figure 5(b).

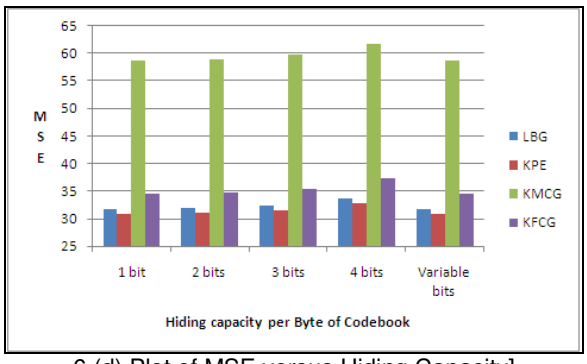


6 (a) Original Cover image Panda.bmp

6 (b) Secret Message



6 (c) Reconstructed image using KPE for Variable bits Method



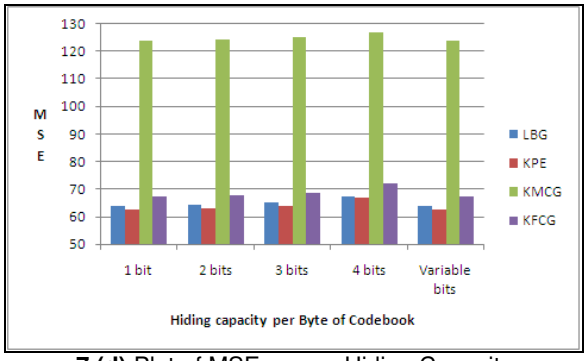

6 (d) Plot of MSE versus Hiding Capacity]

**FIGURE 6:** Results of 1bit, 2bits, 3bits, 4bits and Variable bits on the cover image Panda and secrete image shown in Figure 6(b).

7 (a) Original Cover image Flowers.bmp

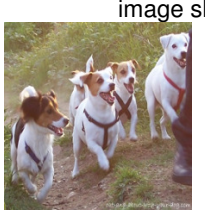
7 (b) Secret Message

7 (c) Reconstructed image using KPE for Variable bits Method




**7 (d)** Plot of MSE versus Hiding Capacity

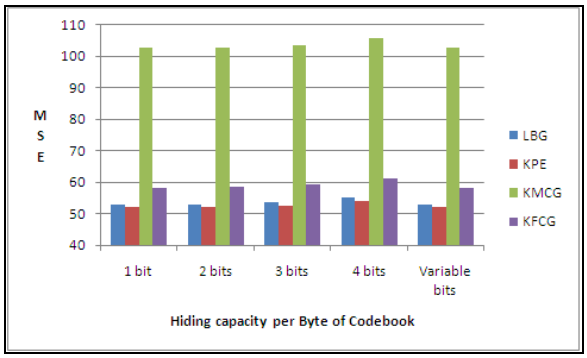
**FIGURE 7:** Results of 1bit, 2bits, 3bits, 4bits and Variable bits on the cover image Flowers and secrete image shown in Figure 7(b).



8 (a) Original Cover image Manydogs.bmp

8 (b) Secret Message

8(c) Reconstructed image using KPE for Variable bits Method

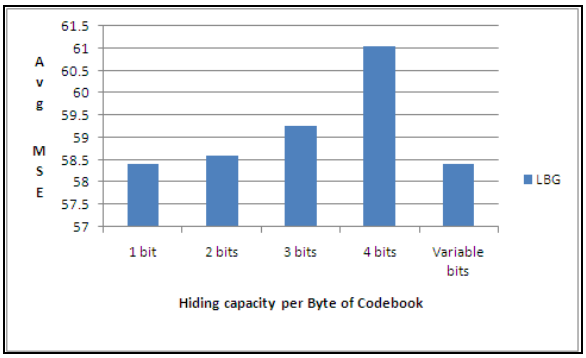



8 (d) Plot of MSE versus Hiding Capacity

**FIGURE. 8:** Results of 1bit, 2bits, 3bits , 4bits and Variable bits on the cover image Manydogs and secrete image shown in Fig. 8(b).

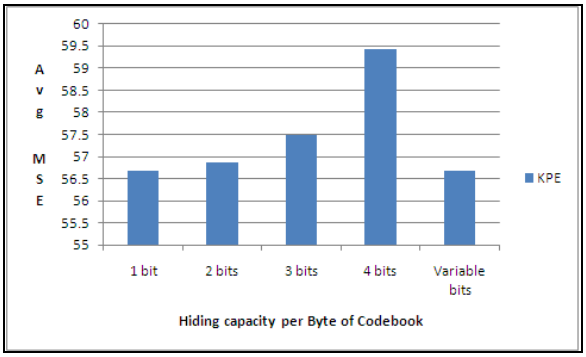
It is observed from Figure 4 to Figure 8 that KPE algorithm gives less MSE in all the data hiding methods 1bit, 2bits, 3bits, 4bits and variable bits as compared to LBG, KMCG and KFCG codebook. LBG and KPE results are comparable.

Figure 9 to 12 shows the results of avg mse versus hiding capacity for various codebook generation techniques by taking average of MSEs for 1 bit, 2 bits, 3 bits, 4 bits and variable bits hiding methods on the various cover images Eleafrica, Bird, Panda, Flowers and Manydogs hiding same secrete image for fair comparison respectively.



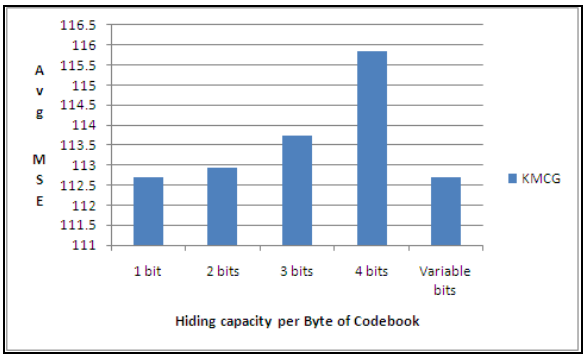

Plot of Avg. MSE versus Hiding Capacity

**FIGURE 9:** Plot of Hiding Capacity versus average MSE for various hiding methods 1bit, 2bits, 3bits, 4bits and Variable bits on LBG.



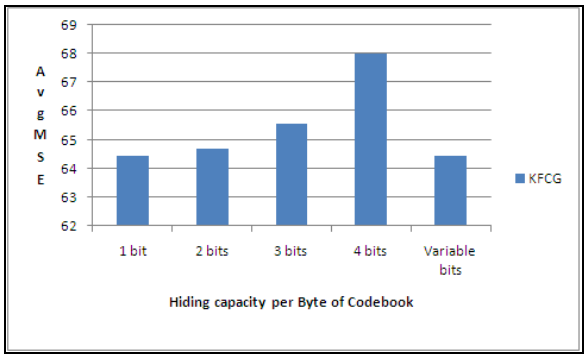

Plot of Avg. MSE versus Hiding Capacity

**FIGURE 10:** Plot of Hiding Capacity versus average MSE for various hiding methods 1bit, 2bits,3bits, 4bits and Variable bits on KPE.




Plot of Avg. MSE versus Hiding Capacity

**FIGURE 11:** Plot of Hiding Capacity versus average MSE for various hiding methods 1bit, 2bits,3bits, 4bits and Variable bits on KMCG.




Plot of Avg. MSE versus Hiding Capacity

**FIGURE 12:** Plot of Hiding Capacity versus average MSE for various hiding methods 1bit, 2bits,3bits, 4bits and Variable bits on KFCG.

From Fig. 9 to Fig 12 it has been observed that variable bit hiding method gives better embedding capacity coupled with low distortion.

## 5. CONCLUSION

Cover image size always dependant on secret image size to be embedded but by using the proposed technique cover image size can be made independent of secret image size. As well as in our proposed technique Secret message get spread throughout the image therefore conventional stegaanalysis technique fail to detect secret data. It is been observed that LBG and KPE gives less MSE compared to other algorithms, proving to be better for steganographic application than KFCG & KMCG. But KFCG and KMCG are faster than KPE and LBG. So tremendous amount of time is saved in the whole process because of using KFCG/KMCG at the cost of slightly more distortion in retrieved messages. Variable bit hiding gives lowest distortion in all VQ codebook generation techniques with considerably large secret data hiding capacity as compared to 1 bit, 2 bits, 3 bits and 4 bits hiding techniques.

## 6. REFERENCES

[1] Available at: http://www.internetworldstats.com/stats.htm [Accessed 7 July 2010]

[2] D. Stinson. *"Cryptography: Theory and Practice",* 2nd Edition, Chapman and Hall/ CRC, February 2002

[3] H. B. Kekre, A. Athawale and P. N.Halarnkar. *"Increased Capacity of Information Hiding in LSBs Method for Text and Image".* International Journal of Electrical, Computer and Systems Engineering, 2(4) Available at: http://www.waset.org/ijecse/v2.html.

[4] H. B. Kekre, A. Athawale and P. N. Halarnkar. *"Polynomial Transformation To Improve Capacity Of Cover Image For Information Hiding In Multiple LSBs".* International Journal of Engineering Research and Industrial Applications (IJERIA), Ascent Publications, Pune, II: 2009

[5] H. B. Kekre, A. Athawale and P. N. Halarnkar. *"Performance Evaluation Of Pixel Value Differencing And Kekre's Modified Algorithm For Information Hiding In Images".* ACM International Conference on Advances in Computing, Communication and Control (ICAC3).2009 Available at: http://portal.acm.org/citation.cfm?id=1523103.1523172.

[6] S. D. Lin, C. F. Chen. *"A Robust DCT-based Watermarking for Copyright Protection".* IEEE Transactions on Consumer Electron, 46(3):415-421, 2000

[7] Y.T. Wu, F.Y. Shih. *"Genetic algorithm based methodology for breaking the steganalytic systems"*. IEEE Transactions on Systems, Man and Cybernetics. Part B, 36(1):24-31, 2006

[8] C. C. Chang, C. Y. Lin. *"Reversible Steganography for VQ-compressed Images Using Side Matching and Relocation"*. IEEE Transactions on Information Forensics and Security, 1(4): 493-501, 2006

[9] C. C. Chang, Y. C. Chou, C. Y. Lin. *"Reversible Data Hiding in the VQ-Compressed Domain"*. IEICE Transactions on Information and Systems, E90-D(9):1422-1429, 2007

[10] A. Gersho, R. M. Gray. *"Vector Quantization and Signal Compressio"*, Kluwer Academic Publishers, Boston, MA, (1991)

[11] H. B. Kekre, T. K. Sarode. *"New Fast Improved Codebook generation Algorithm for Color Images using Vector Quantization"*. International Journal of Engineering and Technology, 1(1):67-77, 2008

[12] H. B. Kekre, T. K. Sarode. *"An Efficient Fast Algorithm to Generate Codebook for Vector Quantization"*. First International Conference on Emerging Trends in Engineering and Technology, ICETET-2008, held at Raisoni College of Engineering, Nagpur, India, 2008 Avaliable at: online IEEE Xplore

[13] H. B. Kekre, T. K. Sarode. *"Fast Codebook Generation Algorithm for Color Images using Vector Quantization"*. International Journal of Computer Science and Information Technology, 1(1):7-12, 2009

[14] H. B. Kekre, T. K. Sarode. *"New Fast Improved Codebook Generation Algorithm for Color Images using Vector Quantization"*. International Journal of Engg. & Tech., 1(1):67-77, 2008

[15] R. M. Gray. *"Vector quantization"*. IEEE Acoust., Speech, Signal Process., 1:4–29, 1984

[16] T. Kim. *"Side match and overlap match vector quantizers for images"*. IEEE Trans. Image Process., 1(4):170–185,1992

[17] W. B. Pennebaker and J. L. Mitchell. "The JPEG Still Image Data Compression Standard". New York: Reinhold, 1993

[18] D. S. Taubman and M. W. Marcellin. *"JPEG2000: Image Compression Fundamentals Standards and Practice"*. Norwell, MA: Kluwer, 2002

[19] A. Gersho and R. M. Gray. *"Vector Quantization and Signal Compression"*. Norwell, MA: Kluwer, 1992

[20] Z. N. Li and M. S. Drew. "Fundamentals of Multimedia. Englewood Cliffs". NJ: Prentice-Hall, Oct. 2003

[21] N. M. Nasrabadi, R. King. *"Image coding using vector quantization: A review"*. IEEE Trans. Commun., 36(8):957–971, 1988

[22] C. H. LEE, L. H. CHEN. *"Fast Codeword Search Algorithm for Vector Quantization"*. IEE Proceedings Image Signal Processing 141(3): 1994

[23] H. B. Kekre, T. K. Sarode, B. Raul. *"Color Image Segmentation using Kekre's Fast Codebook Generation Algorithm Based on Energy Ordering Concept"*. ACM International Conference on Advances in Computing, Communication and Control (ICAC3-2009), pp.:

357-362, 2009. Fr. Conceicao Rodrigous College of Engg., Mumbai. Available at: ACM portal.

[24] H. B. Kekre, T. K. Sarode, B. Raul. *"Color Image Segmentation using Kekre's Algorithm for Vector Quantization"*. International Journal of Computer Science (IJCS), 3(4): 287-292, 2008. Available at: http://www.waset.org/ijcs.

[25] H. B. Kekre, T. K. Sarode, B. Raul. *"Color Image Segmentation using Vector Quantization Techniques Based on Energy Ordering Concept"*. International Journal of Computing Science and Communication Technologies (IJCSCT) 1(2):164-171, 2009

[26] H. B. Kekre, T. K. Sarode, B. Raul. *"Color Image Segmentation Using Vector Quantization Techniques"*. Advances in Engineering Science Sect. C (3): 35-42, 2008

[27] H. B. Kekre, T. K. Sarode. *"Speech Data Compression using Vector Quantization"*. WASET International Journal of Computer and Information Science and Engineering (IJCISE), 2(4):251-254, 2008 Available at: http://www.waset.org/ijcise.

[28] H. B. Kekre, Ms. T. K. Sarode, S. D. Thepade. *"Image Retrieval using Color-Texture Features from DCT on VQ Codevectors obtained by Kekre's Fast Codebook Generation"*. ICGST-International Journal on Graphics, Vision and Image Processing (GVIP), 9(5):1-8, 2009. Available at: http://www.icgst.com/gvip/Volume9/Issue5/P1150921752.html.

[29] H. B. Kekre, T. Sarode, S. D. Thepade. *"Color-Texture Feature based Image Retrieval using DCT applied on Kekre's Median Codebook"*. International Journal on Imaging (IJI), 2(A09):55-65, 2009. Available at: www.ceser.res.in/iji.html (ISSN: 0974-0627).

[30] H. B. Kekre, K. Shah, T. K. Sarode, S. D. Thepade. *"Performance Comparison of Vector Quantization Technique – KFCG with LBG, Existing Transforms and PCA for Face Recognition"*. International Journal of Information Retrieval (IJIR), 02(1):64-71, 2009

[31] H. B. Kekre, T. K. Sarode. *"2-level Vector Quantization Method for Codebook Design using Kekre's Median Codebook Generation Algorithm"*. Advances in Computational Sciences and Technology (ACST), ISSN 0973-6107, 2(2):167–178, 2009. Available at: http://www.ripublication.com/Volume/acstv2n2.htm.

[32] H. B. Kekre, T. K. Sarode. *"Multilevel Vector Quantization Method for Codebook Generation"*. International Journal of Engineering Research and Industrial Applications (IJERIA), 2(V):217-235, 2009, ISSN 0974-1518. Available at: http://www.ascent-journals.com/ijeria_contents_Vol2No5.htm.

[33] H. B. Kekre, T. K. Sarode. *"Vector Quantized Codebook Optimization using K-Means"*. International Journal on Computer Science and Engineering (IJCSE) 1(3):283-290, 2009 Available at: http://journals.indexcopernicus.com/abstracted.php?level=4&id_issue=839392

[34] H. B. Kekre, T. K. Sarode. *"Bi-level Vector Quantization Method for Codebook Generation"*. Second International Conference on Emerging Trends in Engineering and Technlogy, at G. H. Raisoni College of Engineering, Nagpur, 2009, this paper will be uploaded online at IEEE Xplore

# Cutting Edge Practices for Secure Software Engineering

**Kanchan Hans** khans@amity.edu
*Amity Institute of Information Technology*
*Amity University, Noida,*
*201301, India*

**Abstract**

Security has become a high priority issue in software engineering. But, it is generally given a side thought. Security features are implemented after engineering the whole software. This paper discusses that security should be implemented right from the inception of software and planned for each phase of SDLC in software Engineering.The paper also suggests recommendations for implementing security at each phase of lifecycle of software. If each phase of the software engineering includes the appropriate security analysis, defenses and countermeasures, it will definitely result in a more robust and reliable software.

**Keywords:** Requirements analysis, security engineering, Design, Secure Software Engineering, Security vulnerabilities, risk analysis

## 1. INTRODUCTION

Security is considered as a very critical issue for software systems.Software is itself a resource and thus must be afforded appropriate security. But security often isn't the highest priority in software engineering. It is often seen as a task that a team performs in the testing phase at the tail end of the software development lifecycle (SDLC), after the developers have completed the code. Security is generally an afterthought i.e. given due attention only after going through all the phases of engineering a software. But, since the new technologies are coming up using networking, distributed capabilities of systems and also popularity of off the shelf components (COTS), security issues have rather become most important [9]. Security should be considered as an integral part of all phases of software development lifecycle. Also it has been observed that if security is implemented right from the inception of software, it saves the economy billions of dollars. Industry is thus in need of a model to examine security and quality requirements in the development stages of the production lifecycle. Therefore, it is rightly said by –Gene Spafford
"Security is like adding brakes to cars. The purpose of brakes is not to stop you: it's to enable you to go fast!"

## 2. PRACTICES FOR SECURE SOFTWARE ENGINEERING

Software that is developed with security in mind is typically more resistant to both intentional attack and unintentional failures. [10] However, it is incredibly hard to show that a particular system is 100% secure. Presently, there is no single solution for secure software engineering. However, there are specific approaches which improve the likelihood that a system is secure. The security of software is threatened at various points throughout its life cycle, both by inadvertent and intentional choices and actions taken by "insiders"—individuals closely affiliated with the

organization that is producing, deploying, operating, or maintaining the software. Both research and real-world experience indicate that correcting weaknesses and vulnerabilities as early as possible in the software's life cycle is far more cost-effective over the lifetime of the software than developing and releasing frequent security patches for deployed software. If Security mechanisms are fitted into a pre-existing design at a later stage, it will lead to design challenges that usually translate into software vulnerabilities [14]. Therefore security is a necessary property from the beginning of the system's life cycle (i.e., needs and requirements definition) to its end (retirement). Most approaches in practice today encompass training for developers, testers, and architects, analysis and auditing of software artifacts etc. The following section discusses the various security measures to be taken at each phase of SDLC in software engineering.

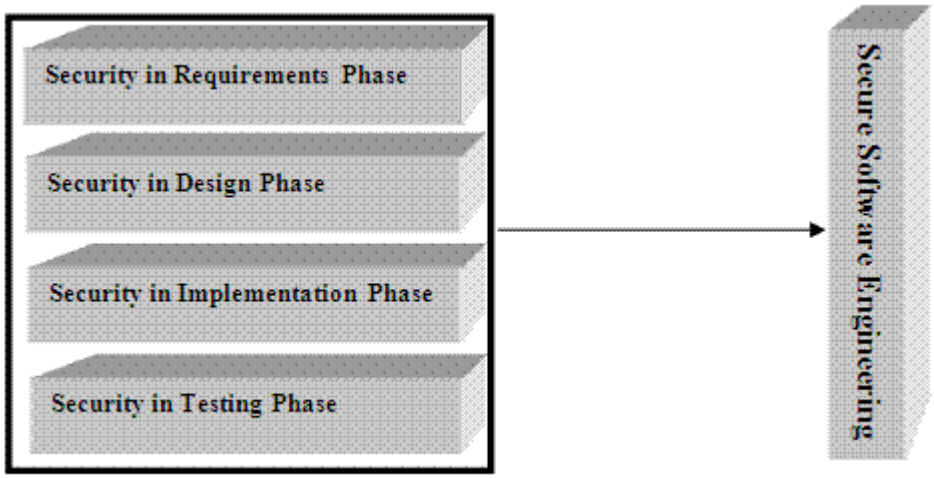



**FIGURE 1:** Secure Software Engineering

**2.1 Security in Requirements Phase**

Building a 100% secure system is hardly possible. In SDLC lot of problems arise because of inadequate Requirements Analysis. It is one of the main causes of over budgeted and delayed projects. Also problems at this phase cause poor quality applications and have reduced scope [1]. So, this phase should be given the utmost importance. From this layer itself, security features should be planned. Detailed requirements of a specific system with respect to security policy should be specified. Planning for such features and adding them at a later point in the life cycle makes this task a lot more difficult. A recent study found that the return on investment when security analysis and secure engineering practices are introduced early in the development cycle, ranges from 12 to 21 percent, with the highest rate of return occurring when the analysis is performed during the application and design.

Recommendations

- Policy on disclosure of information
- Authentication and password management-Use strong passwords. Support password expiration periods and account disablement. Do not store credentials (use one-way hashes with salt). Encrypt communication channels to protect. [13]
- Authorization and role management-Use least privileged accounts. Consider authorization granularity. Enforce separation of privileges. Restrict user access to system level resources.
- Network and data security- Encrypt sensitive data over the wire. Secure the communication channel. Provide strong access controls for sensitive data stores. Code integrity and validation testing.

- Cryptography and key management. Use strong passwords. Support password expiration periods and account disablement. Do not store credentials .Encrypt communication channels to protect.
- Ongoing education and awareness. This involves educating software engineers on general security concepts. Cases on previous security breaches and their consequences should be presented to them in order to appreciate the need for adequate protection of software products. [12]
- Requirements specification review/inspection to find security errors by possibly using a checklist of potential requirements specification security errors.[11]
- Build the abuse *cases* [6]. Similar to use cases, abuse cases describe the system's behavior under attack and building them requires explicit coverage of what should be protected, from whom, and for how long.

**2.2 Security in Design Phase**

At the design and architecture level, a system must be coherent and present a unified security architecture that takes into account security. Designers, architects, and analysts must clearly document assumptions and identify possible attacks.

Recommendations

- Risk should be covered using multiple defensive strategies. In case one layer of protection turns out to inadequate, the next level of defensive strategy will prevent a full breach.
- Secure design guidelines and principles should be followed while developing the initial design. Secure design patterns should either be followed or used for guidance. Secure design decisions should be specified using a secure design specification language.[11]
- The system should be divided into sub parts so that amount of damage that can be done to a system when a unit is compromised [4].
- External review (outside the design team) is often necessary to identify security errors. [8].
- Threat Analysis (risk analysis) should be undertaken that helps identifying issues before code is committed so that they can be mitigated in early SDLC. It involves identifying the conditions that cause incidents and analyzing them. [15]
- Brainstorming discussions should be conducted.
- Standard proven security toolkits (cryptographic and protocol libraries) should be selected.

**2.3 Security in Implementation Phase**

For the implementation phase, a secure programming language should be selected to minimize security errors. Moreover, secure coding standards and guidelines should be followed. [11] This phase also requires full consideration as per security is concerned. Here, the focus must be on implementation flaws. One may make use of tools that scan source code and detect common vulnerabilities. A security team should perform a code walkthrough with the developers and, in some cases, the system architects. A code walkthrough is a high-level walkthrough of the code where the developers can explain the logic and flow. It lets the code review team obtain a general understanding of the code, and it lets the developers explain why certain things were developed the way they were. Some more efforts that can be done are:

Recommendations

- Use of unsafe functions should be avoided or minimized. Look for potential buffer overflows, array out of bound errors, integer underflow and overflow, as well as data truncation errors. [13]

- Latest compiler toolset should be used [6].
- Manual code review must be done. Identify malicious behavior. Know what good traffic looks like.
- All inputs and outputs must be validated. Do not trust input; consider centralized input validation. Do not rely on client-side validation. Be careful with canonicalization issues. Constrain, reject, and sanitize input. Validate for type, length, format, and range. [13]
- Some logging and tracing activity should be followed. Audit and log activity through all of the application tiers. Secure access to log files. Back up and regularly analyze log files.

- Static and dynamic code analysis tools can be used to aid code review process to find vulnerabilities. Static analysis tool, also called source code analyzer examine a program's text without attempting to execute it. Dynamic analysis requires actually running the code. [16]

- Make no assumption: don't suppose something would never happen [9].

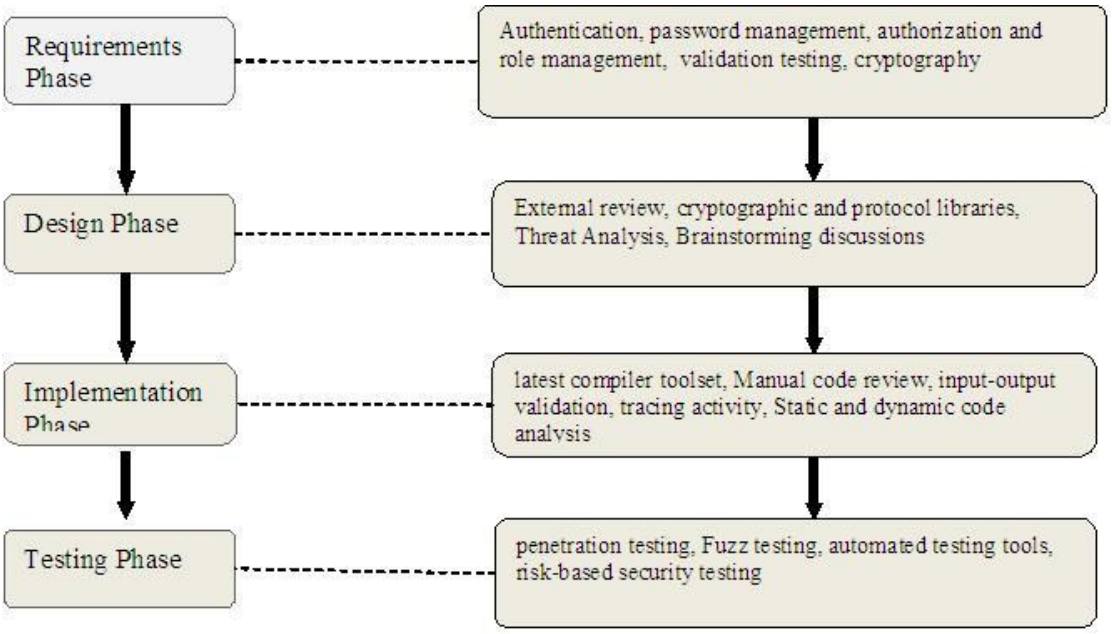



**FIGURE 2:** Integrating Security into each phase of Software Engineering

**2.4 Security in Testing Phase**

During this phase, various tests are conducted to check the security aspect of software being built. A good testing program engages a security team that is predominantly made up of the development team itself. Testing for security involves various techniques.

Recommendations

- Risk-based security testing based on attack patterns and threat models. It involves creating security abuse/misuse cases, performing architectural risk analysis risk-based security test plans. [17]

- Testing security functionality with standard functional testing techniques. Security test plans are prepared for both the strategies.
- Penetration testing should be done. While the application may be secure, a small aspect of the configuration could still be at a default install stage and vulnerable to exploitation. This is where automated black-box scanners are actually effective, looking for the known holes through poor configuration management [3] [7]. Testing most often involves running a series of dynamic functional tests to ensure proper implementation of the application's features. [14]
- Fuzz testing should also be done that provides invalid, unexpected, or random data to the inputs of a program. If the program fails (for example, by crashing or failing) the defects can be noted [2].
- Operations people should carefully monitor fielded systems during use for *security breaks*. So monitoring software behavior is an excellent defensive technique [5].
- Usage of automated testing tools is also recommended.

As risks can emerge up during all stages of the software life cycle, so a constant *risk analysis* thread, with continual risk tracking and monitoring activities, is highly recommended. More important, software development staff on critical software security issues [5].

## 3. CONCLUSION

Hence it is concluded that security must be integrated throughout the software development lifecycle (SDLC) in order to provide the user community with the best, most secure software. We must not leave security requirements to be dealt with as an afterthought. It is needed to incorporate security engineering throughout the software life cycle. If each phase of the software engineering includes the appropriate security analysis, defenses and countermeasures, it will definitely result in a more robust and reliable software. Therefore, security is rightly considered as life blood of software engineering

## 4. REFERENCES

[1] Nancy R. Mead, T. Stehney. *"Security Quality Requirements Engineering (SQUARE) Methodology"*. Software Engineering for Secure Systems -- Building Trustworthy Applications (SESS'05), 2005

[2] Fuzz Testing [Online]. Available at: http://en.wikipedia.org/wiki/Fuzz_testing

[3] Penetration test [Online]. Available at: http://en.wikipedia.org/wiki/Penetration_testing

[4] Jian Chen. *"Security Engineering for Software".* isis.poly.edu/courses/cs996-management /Lectures/SES.pdf

[5] G. McGraw. *"Software Security, Building Security"*. In published by IEEE Computer Society, 2004

[6] G. Blitz, Jarry, M. Coles, Dhillon, C. Fagan. *"Fundamental Practices for Secure Software Development: A guide to most effective secure practices today".* Safe Code Software Forum for Excellence in Code, 2008

[7] G. McGraw. *"Testing for Security during Development: Why We Should Scrap Penetrate-and-Patch".* IEEE Aerospace and Electronic Systems, 13(4):13–15, 1998

[8] G. McGraw. *"Building Secure Software: Better than Protecting Bad Software ".* IEEE Software, 19(6):57–59, 2002

[9] D. J. Hulme, B. Wassermann. *"Software Engineering for Security".* Available at: www.cs.ucl.ac.uk/ staff/ucacwxe/lectures/3C05-01-02/aswe17.pdf

[10] Allen, Julia, Barnum, Sean, Ellison, Robert, McGraw, Gary, Mead, Nancy. *"Software Security Engineering: A Guide for Project Managers".* Addison-Wesley, 2008

[11] M. U. A. Khan, M. Zulkernine. "*A Survey on Requirements and Design Methods for Secure Software Development"*. Technical Report No. 2009 – 562 , School of Computing, Queen's University, Kingston, Ontario, Canada, 2009

[12] Sodiya, Onashoga, Ajayi. *"Towards Building Secure Software Systems, Issues in Informing Science and Information Technology"* . 3: 2006

[13] J. D. Meier, A. Mackman, B. Wastell, P. Bansode, J. Taylor, R. Araujo. "Software *Engineering Explained: Patterns and Practices"*. Microsoft

[14] G. McGraw. *"Software Penetration Testing, Building Security In".* published by IEEE Computer Society, 2005

[15] Barbato, A. Montes, Vijaykumar. *"Methodologies and Tools for Software Vulnerabilities Identification"*

[16] G. McGraw. ***"****Automated Code Review Tools Used for Security, How Things WorK".* Cigital, 2005

[17] G. McGraw. *"Software Security Testing, Building Security In"*. published by IEEE Computer Society, 2004

# Remodeling of Elliptic Curve Cryptography Scalar Multiplication Architecture using Parallel Jacobian Coordinate System

**Adnan Abdul-Aziz Gutub** aagutub@uqu.edu.sa
*Center of Excellence in Hajj and Omrah Research,*
*Umm Al-Qura University P.O. Box: 6287,*
*Makkah 21955, Saudi Arabia*

**Abstract**

In this paper, an improved parallel elliptic curve processor is designed and modeled. We adjusted the Jacobian coordinates system by interacting point double and point add operations. This modified coordinates is parallelized using four multipliers similar to older parallel architectures. We implemented the components of the proposed design using FPGA with parametric features, in terms of number of parallel multipliers, number of parallel adders and width of input operands. The remodeled design is compared to other similar designs i.e. parallel Jacobian coordinates and parallel standard projective coordinates yielding better performance. Results showed that this proposed modified Jacobian design gave higher speed and cost ($AT^2$) showing attractive research direction.

**Keywords:** Cryptography hardware, Elliptic curve cryptography; Jacobian coordinate system; Parallel multipliers architecture; Projective coordinate cryptosystems

## 1. INTRODUCTION

Elliptic curves were first proposed as a basis for public key cryptography in the mid 1980s independently by Koblitz [1] and Miller [2]. Elliptic curve cryptography (ECC) algorithm is practical than existing security algorithms [3,4]. Because of this fact, it showed real attraction to portable devices (handheld devices) manufacturers and the security of their systems. In fact, through these devices, any one can access either email, or do bank transaction or buy any thing on internet using credit cards with high security standards. Elliptic curve algorithm is promising to be the best choice of these handhelds or similar devices because of low computing power (low battery consumption) and fast execution. ECC further gives very high security as compared to similar crypto systems with less size of key. For example, 160 bit ECC system is believed to provide same level of security as 1024 bit RSA [5,6]. Also, the rate at which ECC key sizes increase in order to obtain increased security is much slower than the rate at which integer based discrete logarithm (DL) or RSA key sizes increase for the same level increase in security [7].

Elliptic curves provide a public key crypto-system based on the difficulty of the elliptic curve discrete logarithm problem, which is so called because of its similarity to the discrete logarithm problem (DLP) over the integers modulo a prime p [3,4,8]. This similarity means that most cryptographic procedures carried out using a cryptosystem based on the DLP over the integers modulo p can also be carried out in an elliptic curve cryptosystem. ECCs can also provide a faster implementation than RSA or DL systems, and use less bandwidth and power [9]. These issues are crucial in lightweight applications, i.e. smart cards [10].

An elliptic curve over a Galois field with p elements, GF(p), where p is prime and p > 3 may be defined as the points (x,y) satisfying the curve equation $E: y^2=x^3+ax+b \ (mod\ p)$ , where a and b are constants satisfying $4a^3+27b^2 \neq 0 \ (mod\ p)$. In addition to the points satisfying the curve equation E, a point at infinity ($\phi$) is also defined. With a suitable definition of addition and doubling of points [2], this enables the points of an elliptic curve to form a group with addition and doubling of points being the group operation, and the point at infinity being the identity element. We then further define scalar multiplication of a point P by a scalar k as being the result of adding the point P to itself k times (i.e. kP = P + P + P + · · · + P (k- times)). The elliptic curve discrete logarithm problem is then defined as to compute scalar k such that Q = kP; given the prime modulus p, the curve constants a and b, and two points P and Q. This problem is infeasible for secure elliptic curves [1,2], and thus scalar multiplication is the basic cryptographic operation of an elliptic curve. Scalar multiplication involves mainly three modular operations: addition, multiplication and inversion, where the modular addition operation is the simplest and least to be worried about [11].

The other two ECC scalar multiplication modular operations are inversion and multiplication. Inversion is known to be the complex and very expensive operation [7], its cost is reduced by converting the normal (x,y) affine coordinate system to projective coordinate system (X,Y,Z), which will add-up more modular multiplications to the process; i.e. it will increase the number of multiplications in both ECC point doubling and adding processes operations to reduce the inversion complexity. Thus, modular multiplication is considered to be the repetitive arithmetic ECC scalar multiplication operation to be focused on.

This work extends our previous work of parallelizing the modular multiplications operations within the elliptic curve scalar multiplication process using four modular multipliers. We remodeled the Jacobian coordinate system by interacting the two ECC point doubling and adding processes. We implemented the components of the proposed parallel scalar multiplication design using Field Programmable Gate Arrays (FPGA) with parametric features, in terms of number of parallel multipliers, number of parallel adders, and width of input operands. The results compared timing of the modular multiplication (digit serial), modular adder and total time needed to perform scalar multiplication for three designs i.e. parallel Jacobian coordinates, parallel standard projective coordinates, and this proposed remodeled hardware designs. Analysis showed that this proposed enhanced Jacobian model via four parallel multipliers and two adders gave better $AT^2$ cost than existing scalar multiplication designs.

The flow or the rest of the paper is as follows. The next section will give a short overview of several attempts and hardware ECC designs related to this work. Section 3 describes the ECC scalar multiplication algorithm in some details. Then, the ECC affine and projective coordinates systems are described in Section 4. In Section 5, we describe our proposed design that we extend in this paper. Sections 6 and 7 are for describing our proposed design for scalar multiplication in ECC and its components FPGA implementations. For displaying the results and analysis, we reserved Section 8. Finally, we concluded with the achievements remarks and future work in Section 9.

## 2. RELATED WORK

Several hardware implementations to compute ECC scalar multiplication have been reported in the literature. Every technique has its pros and cons and requires fitting based on the application need. Many designs were dedicated for $GF(2^m)$ computation since it does not suffer the carry propagation problem. For example, in 1993, Agnew et al. [12] implemented ECC over $GF(2^{155})$ normal basis finite field to be simple and gain efficient solution through an optimal multiplier. Their design used a programmable control processor that achieved high performance but limited to the finite field it is designed for. In 1998, Rosner [13] worked on his thesis to develop a reconfigurable ECC crypto engine. His thesis hardware was dedicated for Galois fields $GF(2^n)$m in standard

base representations implemented using FPGAs. His work proved that a full point multiplication on ECC can be implemented on FPGAs although it is built for $GF(2^n)^m$.

In 2000, Torii and Yokoyama [6] used efficient hardware techniques to implement ECC on a digital signal processor (DSP). Their techniques improved modular multiplications based on Montgomery's multiplication method [14] but specified for pipeline processing on DSP. They devised an improved method for computing the number of multiplications and additions which enhanced computing the point doubling operation. Their ideas have been interesting but restricted to their targeted DSP hardware. In the same year, Bednara et al. [15] presented a focus on field multiplications hardware analysis for ECC FPGA hardware implementation. They analyzed Montgomery field multipliers utilizing lookup tables to gain more efficiency. Their study compared Massey-Omura multipliers with LFSR in terms of area and speed. They evaluated different curve coordinate representations with respect to the number of operations within the fields. The best coordinate system matching their FPGA design was reported.

In 2004, Saqib et al. [16] described a parallel architecture for Computing Scalar Multiplication using Hessian Elliptic Curves over $F(2^{191})$ on FPGA. The design aimed to be parallel in all levels and as general as possible without assuming any hardware type to gain the best possible speed. Their results have been interesting for $GF(2^m)$ parallel architecture. A year later, in 2005, Dyke and Langendoefer [17] implemented ECC using Karatsuba's method. Their implementation used iterative hardware accelerator for polynomial multiplication with extended Galois fields (GF), which resulted in reducing the area consumption for recursive applications. Their approach reduces the energy consumption to 60% of the original approach. However, cost for all this achievement is the increased execution time. In 2006, Al-Somani and Ibrahim [18] proposed high performance $GF(2^m)$ Elliptic Curve Crypto processor that is based on standard representation and uses three multipliers to perform parallel field multiplications. They used mixed coordinate systems in point operations to increase the performance. Their results showed better time complexity than existing designs by 76% when implemented on FPGA for $GF(2^{173})$. Al-Somani et al. in [19] further implements another ECC coprocessor using two multipliers only, with similar ideas that gained good results too. In 2007, Fan et al. [20] proposed parallel computing architecture for ECC scalar multiplication by using two-dimensional parallelism. They improved the performance by 26% and 32%, exploiting only vertical or horizontal parallelism, respectively. Different projective coordinates and recoding the scalar with Non-Adjacent Format (NAF) [14] represented further improvements in the performance with similar ideas. Since we focus our work in this paper on GF(p) ECC, the GF(2) ECC arithmetic and hardware implementations is less concentrated on.

Several ECC hardware designs were introduced for GF(p) scalar multiplications, for example, in 2001, Orlando and Paar [21, 22] proposed an architecture for computation of point multiplication for the ECC define over GF(p). Their architecture is scalable over area and speed and can easily be implemented on FPGA's. The processor used Montgomery multiplier (MM) for modular multiplications. The MM relied on the pre-computation of frequently used values and on the use of multiple processing engines. In 2003, Ors et al. [23] described a hardware implementation of an arithmetic processor suitable for RSA and ECC, due to the fact that they are the commonly used types of Public Key Cryptography. They implemented their processor efficiently for bit-lengths in a systolic array architecture that consists special operational blocks for all operations, for example Montgomery Modular Multiplication, modular addition/subtraction, EC Point doubling/addition, modular multiplicative inversion, EC point multiplier, projective to affine coordinates conversion and Montgomery to normal representation conversion. Their design is so generic and flexible that suffered engineering inefficiency in its speed, area, and power consumption.

In 2004, we [24] proposed a parallel architecture for GF(p) elliptic curve cryptographic processor. We used several multipliers adopting projective coordinates to reduce the inversion complexity within the ECC point operations. Our parallelization in [24] found that projecting ECC using Homogeneous Projective coordinates gave better results compared to Jacobian coordinates. Later, in 2005, Ansari, and Huapeng [25] introduced separating point addition and point doubling

operations using two parallel processors to compute kP operations (scalar ECC multiplications). They used a buffer to hold results of point doubling while point addition is still in operation. They have shown that their parallel processors methods raised the operations speed by 90% compared to the single processor methods. However, this ratio in [25] is found dependant on the selection of ECC coordinate system and cannot be generalized. Sozzani and Turcato, in 2005 too, [26] proofed that ECC implementation in hardware is much faster than software implementation. They implemented ECC using hardware CMOS technology (VLSI HCMOS9 library, STMicroelectronics) using some level of parallelization which gave some specific improvement. In the same year, Chen et al. [27] presented a concurrent algorithm to speed up the point multiplication for the ECC based cryptosystem. They have used extra memory space to store intermediate points. The proposed algorithm achieved 100% hardware utilization that depends on the presented time schedule. Their work is found improving a 2001 paper [28], which saved around 32.2% delay for 256 bits computation. A year later, in 2006, Mishra [29] proposed pipelining scheme for implementing the ECC. This scheme enhances the computation of scalar multiplication significantly based on accessing a multiplier for each pipe stage. The pipelining scheme works based on a key observation i.e. to start the subsequent operation without waiting for the previous step to complete. The idea is interesting but needs further elaboration, which is taking place as in the work future study.

In 2007, Al-Khaleel et al. [7] introduced a technique for implementing ECC on FPGAs. It based its design on radix-4 modular multipliers, to allow for more efficient bit processing than radix-2. The hardware also exploited parallelism through proper scheduling and mapping of the algorithm. It presented area time tradeoff when adopting partitioning schemes and folded pipeline techniques. Chelton in 2008 [30] carried out a hardware development on ECC through application specific instruction set (ASIP) to gain high-performance using FPGA technology. They developed a combination of point-doubling operation and point addition operations based on the proposal in [31]. For gaining speed in the operation processes, the data path was pipelined allowing different levels of operation parallelism. The study showed the clock frequency increase and the optimal pipeline depth which changes by changing the FPGA platform.

In this work, we propose a remodeled scalar ECC multiplication architecture that extends our research in [24]. We benefited from the work scheme in [25] inverting it. Instead of separating the point addition and point doubling operations, we mix them benefiting from the scheduling thoughts of [7, 30, 31]. We found that Jacobian coordinates is more appropriate when mixing and parallelizing in four multipliers turning around one achievement of [24]. To build a fair study, we implemented the components used in all designs in FPGA to be used for the compareson. We, then, compared this modified mixed ECC Jacobian coordinates structure with similar parallel projective coordinates structures, i.e. original Jacobian coordinates and standard one. The comparison showed promising results that open directions for interesting research.

## 3. SCALAR MULTIPLICATION ALGORITHM

The algorithm used for scalar multiplication is based on the binary method [34], since it is efficient for hardware implementation. The binary method algorithm is shown below:

Inputs: k: a constant , P: point on the elliptic curve
Output: Q: another point on the elliptic curve, Q=k . P
Define: w: number of bits in k, where $k_i$ is the $i^{th}$ bit in k

```
If k_{w-1}:=1 then Q := P else Q := 0;
for i:= w-2 down to 0 do
      Q := Q + Q;    Point Doubling
      If kw-1=1 then  Q := Q + P;    Point Addition
Return Q;
```

Basically, the binary method algorithm scans the bits of the constant k, in our case, from most to least bit and doubles the current point Q each time. After each point double operation, if the current k bit is one, then the algorithm adds the current point Q to the base point P. Each point operation, double or add, involves three elementary operations: modular multiplication, modular addition and modular multiplicative inverse.

Finding multiplicative inverses in the field GF(p) is extremely slow, and is generally avoided as much as possible [7]. The use of coordinate systems other than the Affine coordinate system (will be illustrated later) greatly reduces the number of inversions required in the operations of the scalar multiplication on the expense of extra multiplications.

ECC use effectively point doubling and addition operations in arithmetic execution. From many years of research, optimize formulae are available for the operations. Especially, by eliminating the costly field inversion from the main loop of the scalar multiplication, fast operations is achieved by using projective coordinates [32]. However, as in [33], the operation in projective coordinate involves more scalar multiplication than in affine coordinate and ECC on projective coordinate will be efficient only when the implementation of scalar multiplication is much faster than multiplicative inverse operation. Therefore, transfer is needed from one coordinate to another for avoiding the inversion process cost. The following section is dedicated for illustration of the coordinate systems structure used for these purposes.

## 4. THE COORDINATE SYSTEMS

An elliptic curve can be represented by several coordinate systems [11]. Following are descriptions of three coordinates, i.e. affine coordinate, standard projective coordinate, and Jacobian projective coordinate procedures.

### 4.1 Affine coordinate:

Let E an elliptic curve over GF(p), has the following equation:
**E**: $y^2=x^3+ax+b$ (mod $p$) , where $a$ and $b$ are constants satisfying $4a^3+27b^2 \neq 0 \pmod p$.
Let $P=(x_1,y_1)$, $Q=(x_2,y_2)$, and $P+Q=(x_3,y_3)$, be points of E(GF(p)) ,

Addition formula: $x_3=\lambda^2-x_1-x_2$, $y_3=\lambda(x_1-x_3)-y_1$, where $\lambda=(y_2-y_1)/(x_1-x_2)$
Doubling formula: $x_3=\lambda^2-2x_1$, $y_3=\lambda(x_1-x_3)-y_1$, where $\lambda=(3x_1^2+a)/(2y_1)$

Addition time = 3 M + 6 S + 1 Inversion
Doubling time = 3 M + 4 S + 1 Inversion

### 4.2 Standard Projective coordinate:

For standard projective coordinates, we set x=X/Y and y=Y/Z, giving the equation:

$$Ep : Y^2Z=X^3+aXZ^2+bZ^3 \pmod p ,$$

Let $P=(X_1,Y_1,Z_1)$, $Q=(X_2,Y_2,Z_2)$ and $P+Q=(X_3,Y_3,Z_3)$ be points of E(GF(p)) ,

Addition formula: $X_3= vA$ , $Y_3= u(v^2X_1Z_2 - A) - v^3Y_1Z_2$ , $Z_3=v^3Z_1Z_2$
where, $u = Y_2Z_1 - Y_1Z_2$ , $v = X_2Z_1 - X_1Z_2$ , $A = u^2Z_1Z_2 - v^3 - 2v^2Y_1Z_2$

Doubling formula: $X_3= 2hs$ , $Y_3= w(4B-h) - 8s^2Y_1^2$ , $Z_3=8s^3$
where, $w = aZ_1^2+3X_1^2$, $s = Y_1Z_1$, $B = X_1Y_1s$ , $h = w^2-8B$

Addition time = 12 M + 2 S
Doubling time = 7 M + 5 S

### 4.3 Jacobian coordinate:

For Jacobian coordinates, we set $x=X/Z^2$ and $y= Y/Z^3$, giving the equation:

$$Ep : Y^2=X^3+aXZ^4+bZ^6 (\text{mod } p) ,$$

Let $P=(X_1,Y_1,Z_1)$, $Q=(X_2,Y_2,Z_2)$ and $P+Q=(X_3,Y_3,Z_3)$ be points of $E(GF(p))$ ,

Addition formula: $X_3=-H^3-2U_1H^2+r^2$ , $Y_3=-S_1ZH^3+r(U_1H^2-X_3)$ , $Z_3=HZ_1Z_2$
where, $U_1=X_1Z_2^2$ , $U_2=X_2Z_1^2$ , $S_1=Y_1Z_2^3$ , $S_2=Y_2Z_1^3$ , $H=U_2-U_1$ , $r=S_2-S_1$

Doubling formula: $X_3=T$ , $Y_3=-8Y_1^4+M(S-T)$ , $Z_3=2Y_1Z_1$
where, $S = 4X_1Y_1^2$ , $M=3X_1^2+aZ_1^2$ , $T= -2S+M^2$

Addition time = 12 M + 4 S
Doubling time = 4 M + 6 S

## 5. PREVIOUS DESIGNS

In [24], the two projective coordinates, standard (Section 4.2) and Jacobian (Section 4.3), have been implemented using parallel architecture, as shown in Figure 1. This design uses four modular multipliers to process inputs according to a proposed specific data path. The data path for projective coordinates addition and doubling are shown in Figure 2 and Figure 4, respectively. Similarly, Figure 3 and Figure 5 show respectively, the data path for addition and doubling, when Jacobian coordinates are implemented. Since the addition/subtraction is very fast compared to multiplication, one adder is used in this design.

The two implementations of the coordinates were compared in [24] with respect to their critical paths and their hardware utilization. The study resulted in choosing the standard projective coordinate as the appropriate efficient choice for parallel designs.

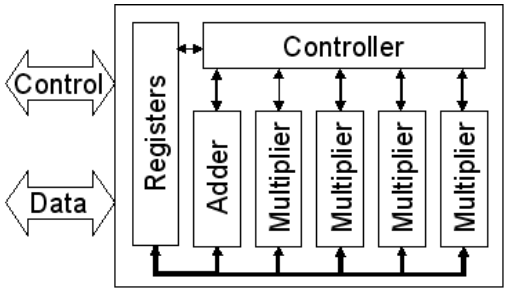



**FIGURE 1:** Elliptic Curve Processor Architecture [1]

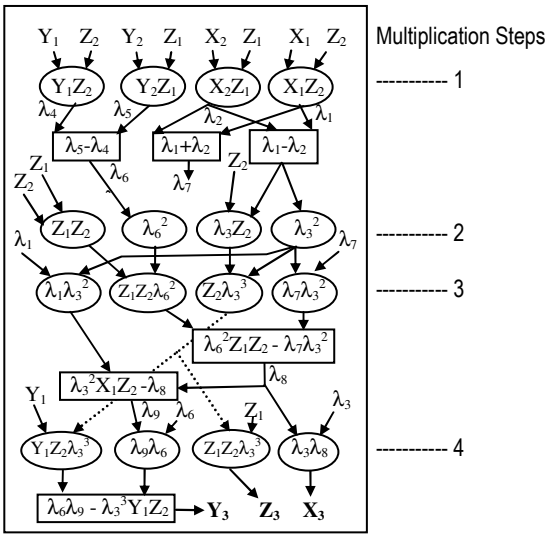



**FIGURE 2**: Projecting *(X,Y)* To *(X/Z,Y/Z)* Adding Two Points Data Flow

## 6. PROPOSED DESIGN

The work proposed here found some interesting benefit from improving the Jacobian coordinate system for parallel hardware designing. Since ECC point add operation is not needed all times, i.e. required based on the value of scalar k (Section 3); and this add operation involves more modular multiplications than point double, we propose to transform some of the modular multiplications needed by the point addition procedure to be pre-computed within the point doubling phase. The idea also makes the most usage of the hardware by allowing the multipliers not used in the last stages of Jacobian point double operation (Figure 5) to be fully utilized. In fact, the hardware of Figure 1 is modified by proper scheduling and adding one more adder makes the remodeled point doubling operation computed in three multiplication times, as shown in Figure 6. The hardware is modified but not much, i.e. the hardware components will be normal with regard to four modular multipliers but adjusted with two modular adders instead of one.

Our idea is to utilize the unused multipliers of the original parallel Jacobian procedure shown in Figure 5, to prepare some pre-multiplications, that may be needed later for the next operation, i.e. may be needed by if point addition operation is required. These pre-multiplications are $U_1=X_3Z_2^2$ and $U_2=X_2Z_3^2$, as shown in Figure 6. As the Jacobian point doubling is rescheduled on three multiplication steps, interestingly the related point adding is rescheduled to be performed in three multiplication times too, as shown in Figure 7.

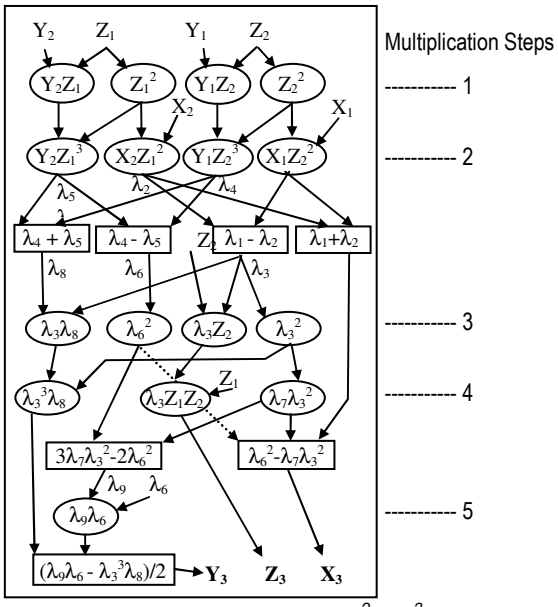



**FIGURE 3**: Jacobian Projecting *(X,Y)* To *(X/Z², Y/Z³)* Adding Points Data Flow

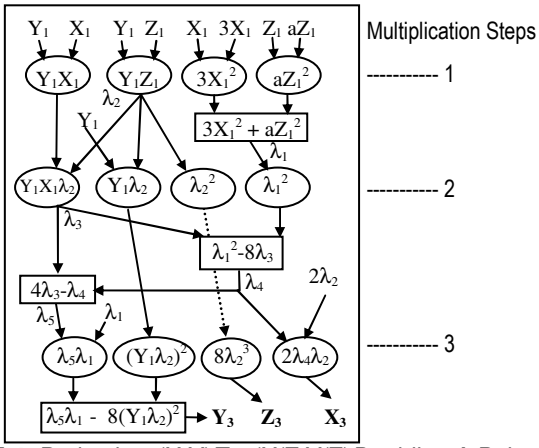



**FIGURE 4:** Projecting *(X,Y)* To *(X/Z,Y/Z)* Doubling A Point Data Flow

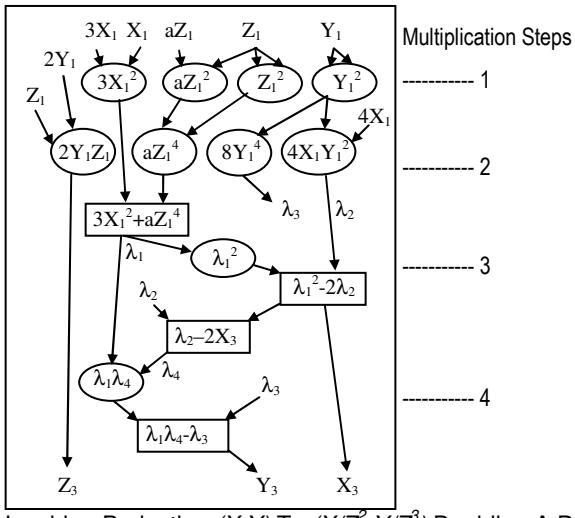



**FIGURE 5**: Jacobian Projecting *(X,Y)* To *($X/Z^2, Y/Z^3$)* Doubling A Point Data Flow

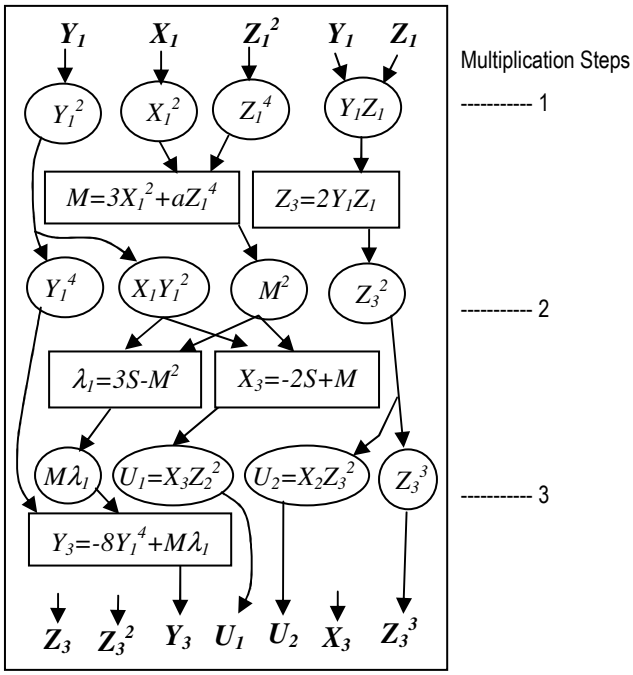



**FIGURE 6**: Proposed Data Flow For Doubling A Projective Point

The top level overview of the new ECC procedure and the complete data transfer is shown in Figure 8. Observe that all second point values ($X_2$, $Y_2$, $Z_2$, $Z_2^2$, $Z_2^3$) are initially not needed by the point doubling making them stored until their tern comes in the point addition. The F condition depends on the scalar value *k* to direct the procedure for point addition operation, if needed. The pre-computed values, $U_1$ and $U_2$, are needed only for point addition operation. If the *k* value directs the function F not to run point addition, the pre-computed values are to be ignored. Some variable are reallocated or reassigned to other registers after point doubling and point adding operations. These are found essential for proper mapping of data within the remodeled procedure to operate correct and efficient.

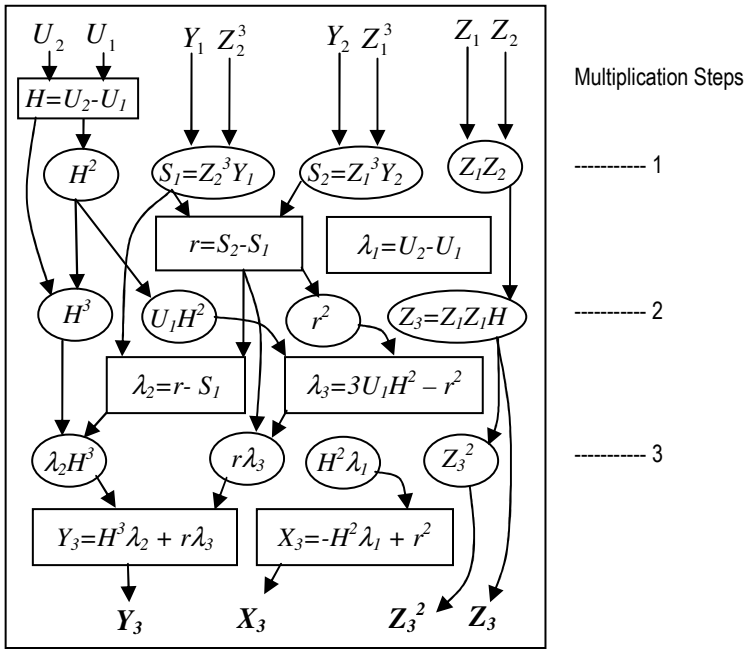



**FIGURE 7**: Proposed Data Flow For Adding a Projective Point

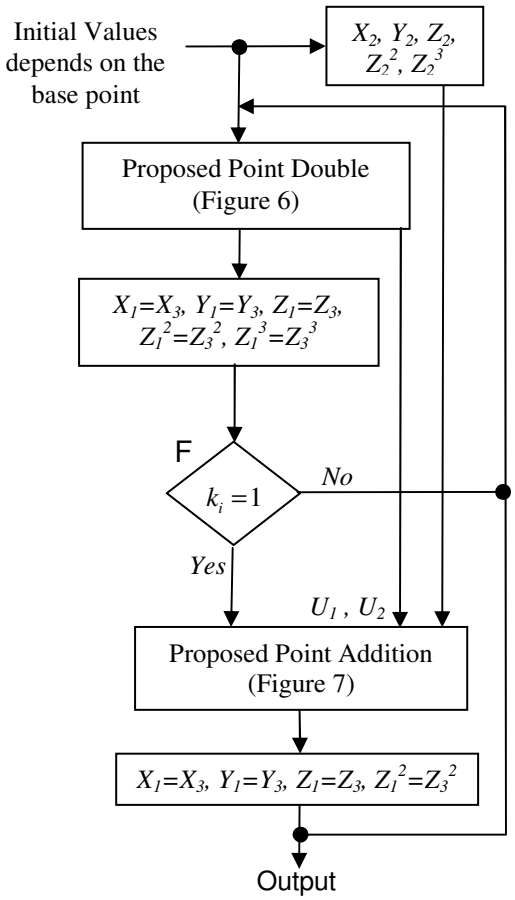



**FIGURE 8:** Overview of Proposed Design Flow

## 7. HARDWARE IMPLEMENTATION

The purpose of the hardware implementation is to give some common platform and fair comparison between our proposed architecture and similar previous designs. The focus in this study is not targeted toward the details of the architecture implementation; instead our aim is to extract the hardware time and area parameters of the main blocks to build a fair comparison study between the designs. Therefore, our implementation exploration here is going to be limited to the level needed to serve this comparison goal.

We will implement the basic blocks of hardware that are commonly used to build all studied designs, i.e. our model here as well as similar previous architectures. The major common components needed by all designs are modular multiplier and modular adder. We described these designs in VHDL and synthesized them for Xilinx Spartan-3 FPGAs. The implementation features of the two basic components are detailed in this section.

### 7.1 Modular Multiplier Implementation:

The modular multiplier is designed to run Montgomery multiplication in binary format, which is proven to be the efficient operation, similar in principle to the work in [21,22,23]. The Montgomery multiplier algorithm is expressed as the following:

$$s_0 = 0$$

$$for\ \ i = 0\ \ to\ \ n\ \ do$$

$$q_i = S_i \mod 2^k$$

$$S_{i+1} = (S_i + q_i \cdot M)/2 + b_i \cdot A$$

$$end$$

The main operation running this Montgomery multiplication is simply modular addition. This made the multiplier algorithm implemented using two cascaded Carry save adders (CSA) connected as shown in Figure 9. We found that these CSA elements are the fastest components in our implemented system; which made the decision of adjusting our system clock. The clock is dominated to the run the signals through two cascaded CSA plus their registering delay. For example, the time needed for an $n \times n$ multiplier to operate is: $T_{mod\_mul} = n \times CLK + 3 \times CLK$; where the additional term: $3 \times CLK$ is to compensate for the final propagate adder. The study assumed that the hardware number of bits used to be *160-bits*, as needed by practical applications of ECC [5,6]. The results showed that for a $160 \times 160$ multiplier running with a clock period of *12ns*, the multiplication can be performed in around *2us*.

The *CLK* period is set to be equal to the longest path delay required by an iteration, which is:
$CLK = T_{iteration} = 2T_{CSA} + T_{REG}$
So, the total delay for an $n \times n$ modular multiplier becomes: $T_{total} = n(2T_{CSA} + T_{REG}) + T_{PCA}$

The multiplier is designed, synthesized and tested through VHDL. It is implemented on FPGA using flexible parameterizble features. The design word size is arranged to be an input parameter (*W*) to the synthesizer. The VHDL code is first complied using value of *W=4*, which is used to build a functional (behavioral) simulation platform. Then, a timing simulation (after place and route) is tested. After that, the design is complied for *W=160-bits*, and the time required for the multiplication result to be ready is computed. The multiplication time found through this process is *2.2usec*, i.e. for our *160-bits* experimentation.

### 7.2 Modular Addition Implementation:

During the ECC point add/double operations (Figures 6 and 7), an extra addition hardware module is needed beside the multipliers. This modular adder involves several hardware units, such as a controller, counter, shifter, Carry Propagate Adder (CPA), accumulator (ACC), and a multipliexor (Mux), connected as shown in Figure 10. Some occasions required the addition

operations to involve adding more than two operands, which lead the adder design to be optimized as follows:

- First, the register of the accumulator (ACC) initializes its results to zero.
- Then, all the required operands are added into ACC.
- Finaly, ACC is reduced to accommodate the modular GF(p) requirement.

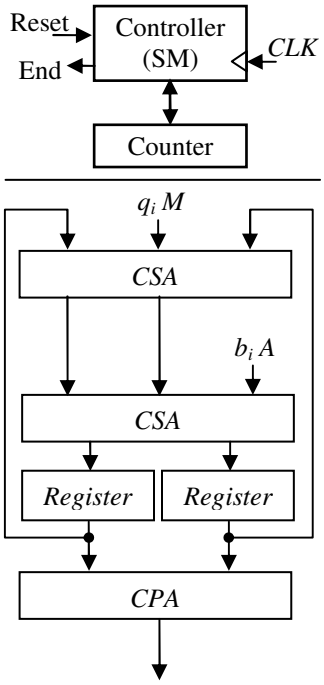



**FIGURE 9:** Block Diagram of the Modular Multiplier Unit

For example, when I want this modular addition unit to compute the following:
$R = (a + 5 \times b) \bmod M$
the following operation sequence are executed:

- ACC = 0 (at this stage registers of ACC are initialized to zero)
- ACC = ACC + *b* (at this stage ACC equals *b*)
- ACC = ACC+ $4 \times b$ (simply *b* shifted by two bits and added to *b* making it equal to *5b*)
- ACC = ACC + *a*
- Reduce ACC (compute ACC *mod M*)

By measuring and analyzing the waveform of the VHDL simulation for the *160-bits* modular adder in relation to the modular multiplication, we found that the time needed by the modular addition stage can be represented as: $T_{mod\_add} = 0.18 \times T_{mod\_mul}$

## 8. COMPARISONS AND ANALYSIS

Using the implemented multiplication and addition units (Section 7) three ECC designs are compared. We compared the proposed design with two existing similar designs are all studied in relation to their area and time, as showed in Table 1. Since the basic components are the same implemented in FPGA, the comparison is believed to be fair and very close to reality. The study considered the area and timing of the internal registers, which cannot be avoided; an *n-bits* CSA is identical to an *n-bits* register. To make our study consistent with the previous study in [24], we assume the basic hardware unit as the multiplier. All other units are quantified relative to this multiplier unit, as follows:

- One *n-bits* Register $\approx 0.13$ $n \times n$ multiplier

- One *n-bits* Adder ≈ 0.40 *n×n* multiplier

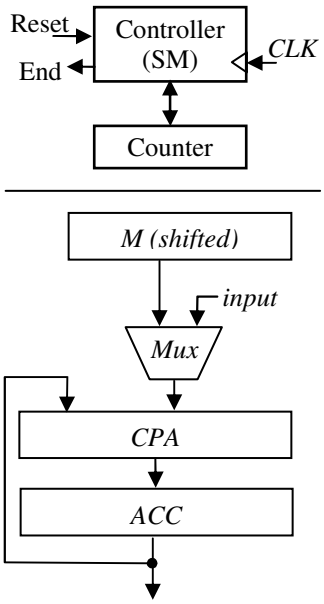



**FIGURE 10:** Block diagram of the modular addition unit

The timing (average cycles multiplication time) comparison counts the addition stages as well as the average multiplication number of cycles (see Table 1); i.e. the multiplication was the only factor in [24] where we added addition timing in this study for more realistic study. The average multiplication number of cycles is computed based on point addition and point doubling according to the ECC binary scalar multiplication algorithm described in Section 3. The multiplication time is computed by adding the total number of multiplications of point doubling procedure plus half the number of multiplications of point addition procedure. The point adding is half the point doubling assuming the value *k* in binary as half ones and half zeros as an average statistical data assumption (Section 3).

| **Design** | | **Sequential Projective[6]** | **Parallel Standard Projective[1]** | **Parallel Jacobian Projective[1]** | **Design (One Adder)** | **Proposed Design (Two Adders)** |
|---|---|---|---|---|---|---|
| **Number of** | **Multipliers** | 1 | 4 | 4 | 4 | 4 |
| | **Adder** | 1 | 1 | 1 | 1 | 2 |
| | **Registers** | 6 | 3+3+6 =12 | 4+4+1 =9 | 5+5+6 =16 | 5+5+6 =16 |
| **Total Area (multiplier size)** | | 2.18 | 5.96 | 5.57 | 6.48 | 6.88 |
| **Average Cycles (Multiplication time)** | | 18.18 | 5+1.26 =6.26 | 6.5+1.35 =7.85 | 4.5+1.53 =6.03 | 4.5+0.9 =5.4 |
| **Cost (different figure of merit values)** | **AT** | 40 | 37 | 44 | 39 | 37 |
| | **$AT^2$** | 721 | 234 | 343 | 236 | 201 |
| | **$A^2T$** | 86.4 | 222 | 244 | 253 | 256 |

**TABLE 1:** Comparison Between Different Designs

In this study, the area is figured by a number related to the number of multipliers, i.e. the adders and registers are given an area estimate relative to the multipliers which totals up to an area factor used for comparison reasons. All architectures are designed for *160 bits* crypto

calculations, which is the common number of bits needed by most applications [5,6].

The area factor and timing average estimate will be multiplied together to generate different cost figures, as in Table 1. These cost figures are just simple figure of merit values to be used for evaluation reasons. For example, the cost AT (AT = A×T), assumes that time and area is having similar balanced importance to the application. When timing is more important, the cost figure of merit AT is assumed to be further multiplied by time T making it $AT^2$ ($AT^2$ = A×T×T). On similar concept but with allowing for the application to have more importance to area than time, we included in this study the cost $A^2T$, where the area is squared multiplied by the timing once. This new $A^2T$ cost is believed to be needed for applications with very limited hardware area such as smart cards and small mobile devices, where area is more important than speed.

Based on the cost values, $AT$, $AT^2$, and $A^2T$, an appropriate design can be preferred. All cost figures for all designs are plotted in Figure 11, with some figures rescaled to fit in the graph. The benefit of the cost comparison is chose the preferred design and not in the cost figure value. Our proposed hardware is showing to have the best cost when time is having more priority over area, i.e. for $AT^2$. When the area and time are having the same importance ($AT$ cost), our proposed design is having similar cost to the parallel standard hardware as our design in [24], which is also the lowest preferred value. Here the proposed design is a bit faster the old design but with the benefit I n speed compensated for in the hardware area.

The sequential hardware is preferred over all parallel designs when area is having more importance than time, as shown in $A^2T$ cost plotting in Figure 11. This remark is also including our proposed design in this study, which can be a drawback of this parallel design whenever the hardware area is important much more than the speed.

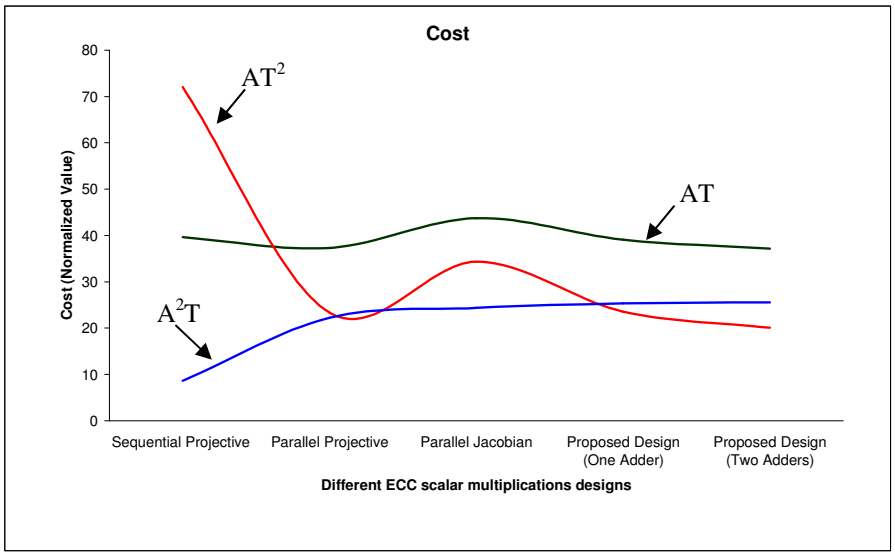



**FIGURE 11:** Different Cost Comparison of All Designs (rescaled to fit in the figure)

## 9. CONCLUSION

This study targeted speeding-up elliptic curve crypto (ECC) computations. We focused on ECC scalar multiplications adopting projective coordinates to reduce the inversion complexity effect. The study proposed remodeling Jacobian projective coordinate system tuned for parallel hardware implementation. We proposed merging ECC point adding and point doubling operations as a new modified method. The proposed hardware is similar to previous designs of four multipliers and an adder but with one more adder making it involve two addition units.

The new architecture is compared to existing scalar multiplication designs. All designs' basic units are implemented similarly on FPGA to insure fair comparison and area time cost analysis. The cost evaluation involved three studies, i.e. $AT$, $AT^2$ and $A^2T$. Our proposed design showed interesting performance results for $AT$ and $AT^2$ costs. The clear improvement is shown in the $AT^2$ cost, where area is not as important as the computation timing. We concluded that implementing the proposed Jacobian coordinate using four multipliers and two adders, yields better $AT^2$ cost than existing scalar multiplication designs. The study is attractive for researchers to observe promising direction behind this research idea.

## ACKNOWLEDGMENTS

The author is grateful to *Professor Mohammad K. Ibrahim* for all his beneficial ideas and engorgements. Thanks to the COE 509: Applied Crypto Systems students, i.e. *Mr Esa Alghonaim* for his contribution in VHDL coding and *Mr Aleem Alvi* for his proofreading and feedback related to this research. Thanks to King Fahd University of Petroleum and Minerals (KFUPM) and Umm Al-Qura University (UQU) for supporting all this work.

## REFERENCES

[1] N. Koblitz, "Elliptic curve cryptosystems", In Mathematics of Computation, volume 48, pages 203–209, 1987.

[2] V. Miller, "Use of elliptic curves in cryptography", *Advances in Cryptology—CRYPTO'85*, Vol. 218 of Lecture Notes in Computer Science, pages 417–426. Springer-Verlag, 1986.

[3] J.H. Cheon, H.J. Kim, S.G. Hahn, "Elliptic curve discrete logarithm and integer factorization", The Math Net Korea, Information Center for Mathematical Sciences (ICMS), February 7, 1999, http://mathnet.kaist.ac.kr/

[4] A Certicom Whitepaper, "The Elliptic Curve Cryptosystem", July 2000, http://www.certicom.com/

[5] Hitchcock, Yvonne Roslyn, "Elliptic Curve Cryptography for Lightweight Applications", *Institution Queensland University of Technology*, 2003. http://adt.library.qut.edu.au/adt-qut/public/adt-QUT20040723.150510/

[6] Naoya Torii and Kazuhiro Yokoyama, "Elliptic Curve Cryptosystem", *FUJITSU Sci. Tech. Journal*, Vol. 36, No. 2, pages 140-146, December 2000. www.fujitsu.com/downloads/MAG/vol36-2/paper05.pdf

[7] O. Al-Khaleel, C. Papachristou, F. Wolff, K. Pekmestzi, "An Elliptic Curve Cryptosystem Design Based on FPGA Pipeline Folding", *13th IEEE International On-Line Testing Symposium, IOLTS 07*, pages 71 – 78, 8-11 July 2007.

[8] A.J. Menezes, T. Okamoto, S.A. Vanstone, S, "Reducing elliptic curve logarithms to logarithms in a finite field", *IEEE Transactions on Information Theory*, Volume 39, Issue 5, pages 1639 – 1646, Sept. 1993.

[9] T. Hasegawa, J. Nakajima, M. Matsui, "A practical implementation of elliptic curve cryptosystems over GF(p) on a 16-bit microcomputer", *In Public Key Cryptography – PKC '98, Proceedings*, volume 1431 of Lecture Notes in Computer Science, pages 182–194, Springer-Verlag, 1998.

[10] Scott Vanstone, "Crypto Column: The Importance of Good Crypto and Security Standards", Code & Cipher- Certicom's Bulletin of Security and Cryptography, Volume 1, Issue 4, 2004, http://www.certicom.com/codeandcipher

[11] A. Daly, W. Marnane, "Efficient Architectures for implementing Montgomery Modular

Multiplication and RSA Modular Exponentiation on Reconfigurable Logic", *Proceedings of the ACM/SIGDA tenth international symposium on Field-programmable gate arrays*, pages: 40 - 49, Monterey, California, USA, 2002

[12] G.B. Agnew, R.C. Mullin, S.A. Vanstone, "An implementation of elliptic curve cryptosystems over $F2^{155}$", *IEEE Journal on Selected Areas in Communications*, Volume 11, Issue 5, pages 804 – 813, June 1993

[13] Martin Christopher Rosner, "Elliptic Curve Cryptosystems on Reconfigurable Hardware", *MS Thesis submitted to Electrical Engineering in Worcester Polytecnic Institute*, U.S.A., 1998.

[14] I. Blake, G. Seroussi, N.P. Smart., "Elliptic Curves in Cryptography", London Mathematical Society, Lecture Note Series. Cambridge University Press, 1999.

[15] M. Bednara, M. Daldrup, J. Teich, J. von zur Gathen, J. Shokrollahi, "Tradeoff analysis of FPGA based elliptic curve cryptography", *IEEE International Symposium on Circuits and Systems, ISCAS 2002*, Vol. 5, pages 797 – 800, 26-29 May 2002.

[16] N.A. Saqib, F. Rodriguez-Henriquez, A. Diaz-Perez, "A Parallel Architecture for Computing Scalar Multiplication on Hessian Elliptic Curves", *International Conference on Information Technology: Coding and Computing (ITCC'04)*, Vol. 2, pages 546–552, Las Vegas, NV, USA, 2004.

[17] Z. Dyka, P. Langendoerfer, "Area efficient hardware implementation of elliptic curve cryptography by iteratively applying Karatsuba's method", *Proceedings of Conference on Design, Automation and Test in Europe*, pages 70 – 75, 2005.

[18] T.F. Al-Somani, M.K. Ibrahim, "High Performance Elliptic Curve GF(2m) Cryptoprocessor Secure Against Timing Attacks", *International Journal of Computer Science and Network Security - IJCSNS*, Vol. 6, No.1B, pages 177-183, January 2006.

[19] T.F. Al-Somani, M. Ibrahim, A. Gutub, "Highly Efficient Elliptic Curve Crypto-Processor with Parallel GF(2m) Field Multipliers", *Journal of Computer Science (JCS)*, Vol. 2, No 5, pages 395-400, 2006.

[20] J. Fan, K. Sakiyama, I. Verbauwhede, "Elliptic Curve Cryptography on Embedded Multicore Systems," *Workshop on Embedded Systems Security - WESS*, pages 17-22, 2007. http://www.cosic.esat.kuleuven.be/publications/article-937.pdf

[21] G. Orlando, C. Paar, "A scalable GF(p) elliptic curve processor architecture for programmable hardware", *Third International Workshop on Cryptographic Hardware and Embedded Systems - CHES*, pages 348-363, Paris, France, 14-16 May 2001.

[22] G. Orlando, "Efficient Elliptic Curve Processor Architectures for Field Programmable Logic", *Ph.D. Thesis, Worcester Polytechnic Institute*, March 2002.

[23] S.B. Ors, L. Batina, B. Preneel, J. Vandewalle, "Hardware implementation of an elliptic curve processor over GF(p)", *IEEE International Conference on Application-Specific Systems, Architectures, and Processors*, pages 433 – 443, 24-26 June 2003.

[24] A. Gutub, M.K. Ibrahim, "High Radix Parallel Architecture For GF(p) Elliptic Curve Processor", *IEEE Conference on Acoustics, Speech, and Signal Processing - ICASSP 2003*, pages 625- 628, Hong Kong, April 6-10, 2003.

[25] B. Ansari, Huapeng Wu,"Parallel scalar multiplication for elliptic curve cryptosystems", *International Conference on Communications, Circuits and Systems*, Vol. 1, pages 71-73, 27-30 May 2005.

[26] F. Sozzani, G. Bertoni, S. Turcato, L. Breveglieri, "A parallelized design for an elliptic curve cryptosystem coprocessor", *International Conference on Information Technology: Coding and Computing - ITCC 2005*, Vol. 1, pages 626 – 630, 4-6 April 2005.

[27] Jun-Hong Chen, Ming-Der Shieh, Chien-Ming Wu,"Concurrent algorithm for high-speed point multiplication in elliptic curve cryptography", *IEEE International Symposium on Circuits and*

*Systems - ISCAS*, pages 5254 – 5257, 23-26 May 2005.

[28] S. Moon, J. Park, and Y. Lee, "Fast VLSI Algorithms for High-security Elliptic Curve Cryptographic Application", *IEEE Trans. on Consumer Electronics*, Vol. 47, No. 3, pages 700-708, 2001.

[29] P.M. Mishra,"Pipelined computation of scalar multiplication in elliptic curve cryptosystems (extended version)", *IEEE Transactions on Computers*, Vol. 55, No. 8, pages 1000 – 1010, Aug. 2006.

[30] W.N. Chelton, M. Benaissa, "Fast Elliptic Curve Cryptography on FPGA", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 16, No. 2, pages 198-205, Feb. 2008.

[31] W. Chelton, M. Benaissa, “High-speed pipelined ECC processor on FPGA”, *IEEE Workshop Signal Process. Syst. (SiPS)*, pages 136-141, Banff, Canada, 2006.

[32] P. Longa, A. Miri, "Fast and Flexible Elliptic Curve Point Arithmetic over Prime Fields", *IEEE Transactions on Computers*, Vol. 57 , No. 3, pages 289-302, March 2008.

[33] M.S. Anoop, "Elliptic Curve Cryptography, An Implementation Guide", online Implementation Tutorial, Tata Elxsi, India, 5 January 2007 http://www.infosecwriters.com/text_resources/pdf/Elliptic_Curve_AnnopMS.pdf

[34] D. Hankerson, A. Menezes, S. Vanstone, “Guide to Elliptic Curve Cryptography”, Springer-Verlag, ISBN 0-387-95273-X, 2004.

# Automated Detection System for SQL Injection Attack

**Dr K.V.N.Sunitha** k.v.n.sunitha@gmail.com
*Professor &Head,*
*Department of Computer Science & Engineering,*
*G.Narayanamma Institute of Technology and Science*
*Shaikpet, Hyderabad – 500 008, A.P., India*

**Mrs.M. Sridevi** sreetech99@gmail.com
*Assoc.Professor,*
*Department of Computer Science & Engineering*
*Laqshya Institute of Technology and Science*
*Konijerla, Khammam – 507 305 , A.P., India*

**Abstract**

Many software systems have evolved as Web-based t that makes them available to the public via the Internet and can expose them to a variety of Web-based attacks. One of these attacks is SQL Injection vulnerability (SQLIV), which can give attackers unrestricted access to the databases that underlie Web applications and has become increasingly frequent and serious. The intent is that Web applications will limit the kinds of queries that can be generated to a safe subset of all possible queries, regardless of what input user provides. SQL Injection attacks are possible due to the design drawbacks of the web sites, which interact with back-end databases. Successful attacks may damage more. We introduce a system that deals with new automated technique for preventing SQL Injection Attacks based on the novel concept of regular expressions is to detect SQL Injection attacks. The proposed system can detect the attacks that are from Internet and Insider Attacks, by analyzing the packets of the network servers.

**Keywords**—Intrusion Detection, Injection Attacks, Regular Expressions, SQL Query.

## 1. INTRODUCTION

Nowadays it is most common for any organization to use database and web application for maintaining their information. Security of these systems became crucial. Internet threats like SQL Injection attacks on database through web applications are more. Solutions for to avoid these attacks are 1. Placing a powerful Network SQL Injection Intrusion Detection Systems (IDS). 2. SQL Injection Insider Misuse Detection Systems(SQLIMDS).

Web applications interface with databases that contain information such as customer names, preferences, credit card numbers, purchase orders, and so on. Web applications build SQL queries to access these databases based, in part, on user-provided input. Inadequate input validation can enable attackers to gain complete access to such databases. One way in which this happens is that attackers can submit input strings that contain specially encoded database commands. When the Web application builds a query by using these strings and submits the query to its underlying database, the attacker's embedded commands are executed by the database and the attack succeeds. The results of these attacks are often disastrous and can range from leaking of sensitive data to the destruction of database contents.

The cause of SQL injection vulnerabilities are relatively simple and well understood: a kind of problem is insufficient validation of user input, developers have proposed a range of coding guidelines that promote defensive coding practices, such as encoding user input and validation. A rigorous and systematic application of these techniques is an effective solution for preventing SQL injection vulnerabilities.

However, in practice, the application of such techniques is human-based and, thus, prone to errors. Furthermore, fixing legacy code-bases that might contain SQL injection vulnerabilities can be an extremely labor-intensive task. Although recently there has been a great deal of attention to the problem of SQL injection vulnerabilities, many proposed solutions fail to address the full scope of the problem. There are many types of SQLIA and countless variations on these basic types. Researchers and practitioners are often unaware of the myriad of different techniques that can be used to perform SQLIA. Therefore, most of the solutions proposed detect or prevent only a subset of the possible SQLIA.

An SQL injection attack (SQLIA) is a type of attack on web applications that exploits the fact that input provided by web clients is directly included in the dynamically generated SQL statements. SQLIA is one of the foremost threats to web applications. According to the SQLIMDS Foundation, injection flaws, particularly SQL injection, were the second most serious web application vulnerability type in 2007. Since they are easy to find and exploit, SQL injection vulnerabilities are frequently employed by attackers.

SQL injection is a technique for maliciously exploiting applications that use client-supplied data in SQL statements. Attackers trick the SQL engine into executing unintended commands via supplying specially crafted string input, thereby gaining unauthorized access to a database in order to view or manipulate restricted data. Using SQLIAs, an attacker may be able to read, modify, or even delete database information. In many cases, this information is confidential or sensitive and its loss can lead to problems such as identity theft and fraud.

In general, SQL injection attacks are a class of code injection attacks that take advantage of the lack of validation of user input. These attacks occur when developers combine hard-coded strings with user-provided input to create dynamic queries. Intuitively, if user input is not properly validated, attackers may be able to change the developer's intended SQL command by inserting new SQL keywords or operators through specially crafted input strings.

We propose a new highly automated approach for dynamic detection of SQL injection attacks. Intuitively, our approach works by identifying "trusted" strings in an application and allowing only these trusted strings to be used to create the semantically relevant parts of a SQL query such as keywords or operators. The general mechanism that we use to implement this approach is based on dynamic tainting, which marks and tracks certain data in a program at runtime. The kind of dynamic tainting that we use gives our approach several important advantages over techniques based on other mechanisms. Our approach is highly automated, does not rely on complex static analyses and is both efficient and precise and, in most cases, requires minimal or no developer intervention. Compared to other existing techniques based on dynamic tainting, our approach makes several conceptual and practical improvements that take advantage of the specific characteristics of SQLIA. The first conceptual advantage of our approach is the use of positive tainting. The second conceptual advantage of our approach is the use of flexible syntax-aware evaluation. The practical advantages of our approach are that it imposes a low overhead on the application and it has minimal deployment requirements. Efficiency is achieved by using a specialized library, called Meta Strings, that accurately and efficiently assigns and tracks trust markings at runtime. The only deployment requirements for our approach are that the Web application must be instrumented and it must be deployed with our Meta Strings library, which is done automatically.

## 2. SQL INJECTION ATTACK APPROACH

Our approach makes several conceptual and practical improvements over traditional dynamic tainting ap proaches by taking advantage of the characteristics of SQLIAs and Web applications. First, unlike existing dynamic tainting techniques, our approach is based on the novel concept of positive tainting, that is, the identification and marking of trusted, instead of untrusted, data. Second, our approach performs accurate and efficient taint propagation by precisely tracking trust markings at the character level. Third, it performs syntax-aware evaluation of query strings before they are

sent to the database and blocks all queries whose non-literal parts (that is, SQL keywords and operators) contain one or more characters without trust markings. Finally, our approach has minimal deployment requirements, which makes it both practical and portable. The following sections discuss these key features of our approach in detail [8].

**A. Positive Tainting VS Negative**

Positive tainting differs from traditional tainting (negative tainting) because it is based on the identification, marking, and tracking of trusted, rather than untrusted, data as shown in Figure 1 . In the context of preventing SQLIAs, the conceptual advantages of positive tainting are especially significant.

With positive tainting, incompleteness may lead to false positives, but it would never result in an SQLIA escaping detection. Moreover, as explained in the following, the false positives generated by our approach, if any, are likely to be detected and easily eliminated early during prerelease testing. Positive tainting uses a white-list, rather than a block list policy.

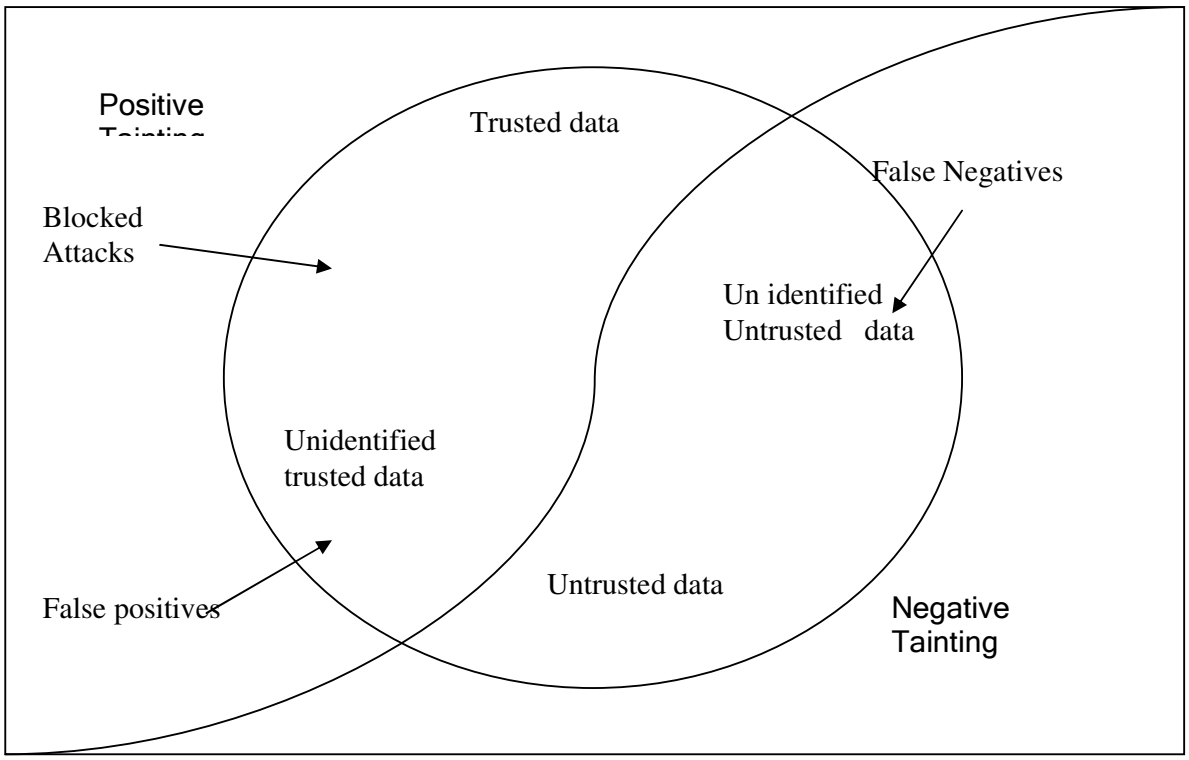



**FIGURE 1**: Positive Tainting vs Negative Tainting.

**B. Accurate and Efficient Taint Propagation**

Taint propagation consists of tracking taint markings associated with the data while the data is used and manipulated at runtime. In our approach, we provide a mechanism to accurately mark and propagate taint information by 1) tracking taint markings at the “right” level of granularity and 2) precisely accounting for the effect of functions that operate on the tainted data [7, 8, 9].

Character-level tainting. We track taint information at the character level rather than at the string level. We do this because, for building SQL queries, strings are constantly broken into sub strings, manipulated, and combined. By associating taint information to single characters, our approach can precisely model the effect of these string operations.

Accounting for string manipulations. To accurately maintain character-level taint information, we must identify all relevant string operations and account for their effect on the taint markings. Our approach achieves this goal by extending all classes and methods (that perform String manipulations), by adding functionality to update taint markings based on the methods’ semantics.

**C. Syntax-aware Evaluation**

Our technique performs syntax-aware evaluation of a query string immediately before the string is sent to the database to be executed. To evaluate the query string, the technique first uses a SQL parser to break the string into a sequence of tokens that correspond to SQL keywords, operators, and literals. The technique then iterates through the tokens and checks whether tokens (i.e., sub strings) other than literals contain only trusted data. If all such tokens pass this check, the query is considered safe and is allowed to execute. If an attack is detected, a developer specified action can be invoked.

Our technique performs syntax-aware evaluation of a query string immediately before the string is sent to the database to be executed. To evaluate the query string with use of regular expressions , this technique first uses a SQL parser to break the string into a sequence of tokens that correspond to SQL keywords, operators, and literals. The technique then iterates through the tokens and checks whether tokens (that is, substrings) other than literals contain only trusted data. If all such tokens pass this check, the query is considered safe and is allowed to execute. If an attack is detected, a developer specified action can be invoked.

The proposed system is a Network Intrusion Detection System (NIDS), will be deployed in between corporate network and the internet. All the packets from corporate network to the internet and the internet to the corporate network will pass through the NIDS,. This system captures the http traffic,. And stores them in a folder called pcaps. This folder is meant for network packet files. Because of the network traffic is mix of http and non-http traffic, the proposed NIDS is meant for the SQL Injection attacks and SQL Injections are web attacks. We filter for the HTTP URLS of the pcaps and extracts them to packet.dat file.

## 3. DEVELOPING REGULAR EXPRESSIONS

The proposed system is a Network Intrusion Detection System (NIDS), will be deployed in between corporate network and the internet. All the packets from corporate network to the internet and the internet to the corporate network will pass through the NIDS,. This system captures the http traffic and stores them for network packet files. The Regular Expression Development Process is Gathering Attack Patterns, Common Regular Expression Development with Regex-Coach, Testing With Regex-Coach. For the attack patterns shown in Figure 2 and Figure 3, such as

‘ or 1=1 -- “
or 123=123 ‘
or ‘a’ = ‘a’ like patterns

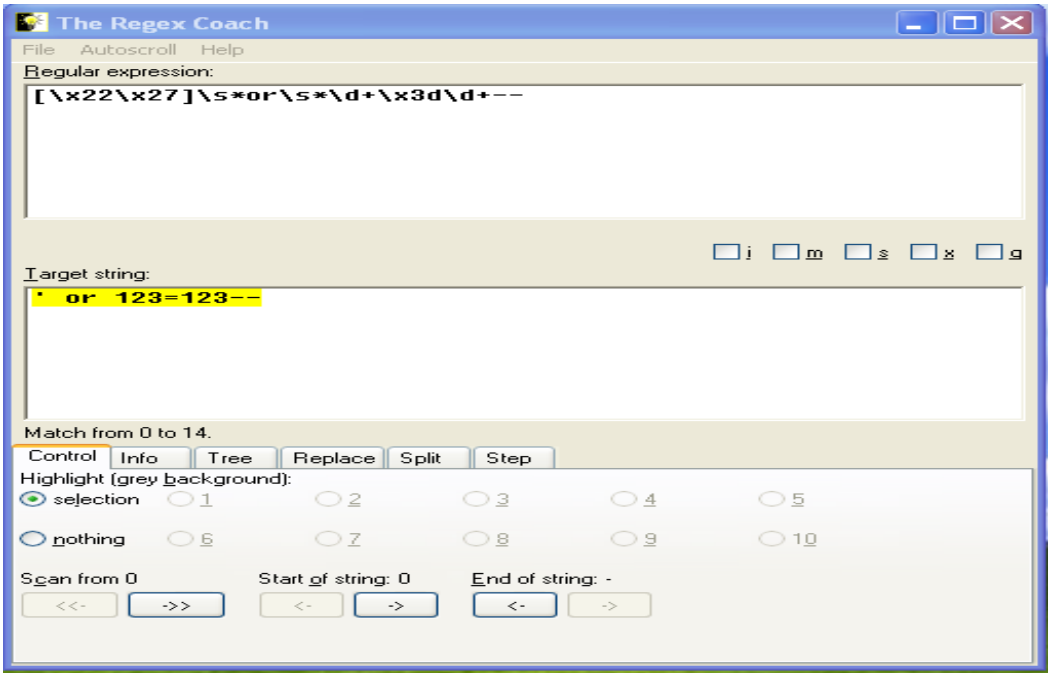



**FIGURE 2**: Regular Expression Will Match With 123=123 Like Patterns.

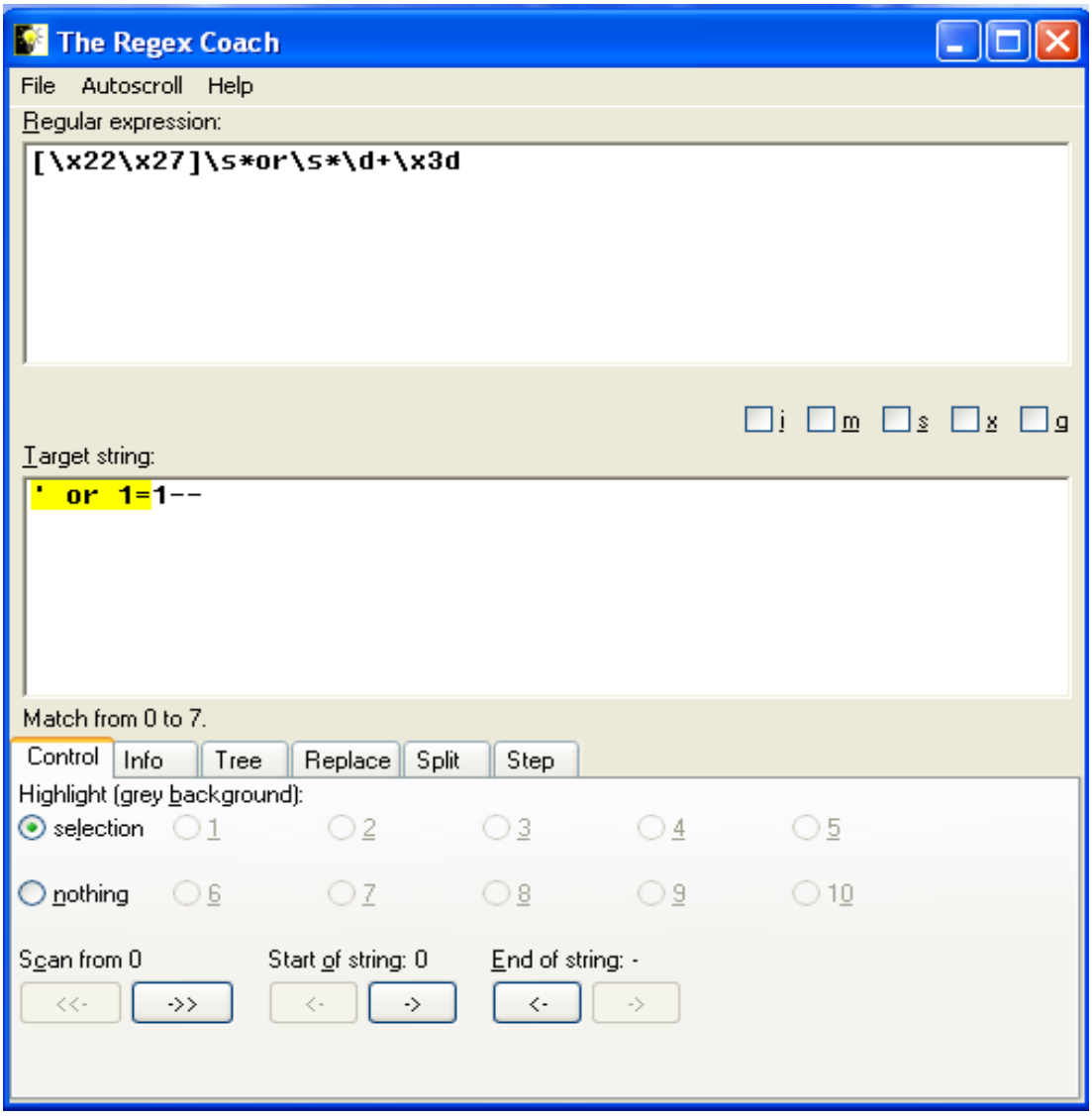



**FIGURE 3**: Regular Expression Will Match With 'Or 1=1 ' Like Patterns

Because of the network traffic is mix of http and non-http traffic, the proposed NIDS is meant for the SQL Injection attacks and SQL Injections are web attacks. After capturing the packets the decoded patterns will be checked against all the regular expressions. If any rule matches, then that log information will be send by the Detection System.

## 4. ALGORITHM DETAILS

Our system has three major steps:

- vulnerability detection,
- preparing the regular expressions,
- report generation.

As per Figure 4. first captures packets from the Ethernet , extract the decoded URL and check with regular expression which already in directory if match ok generate logs.

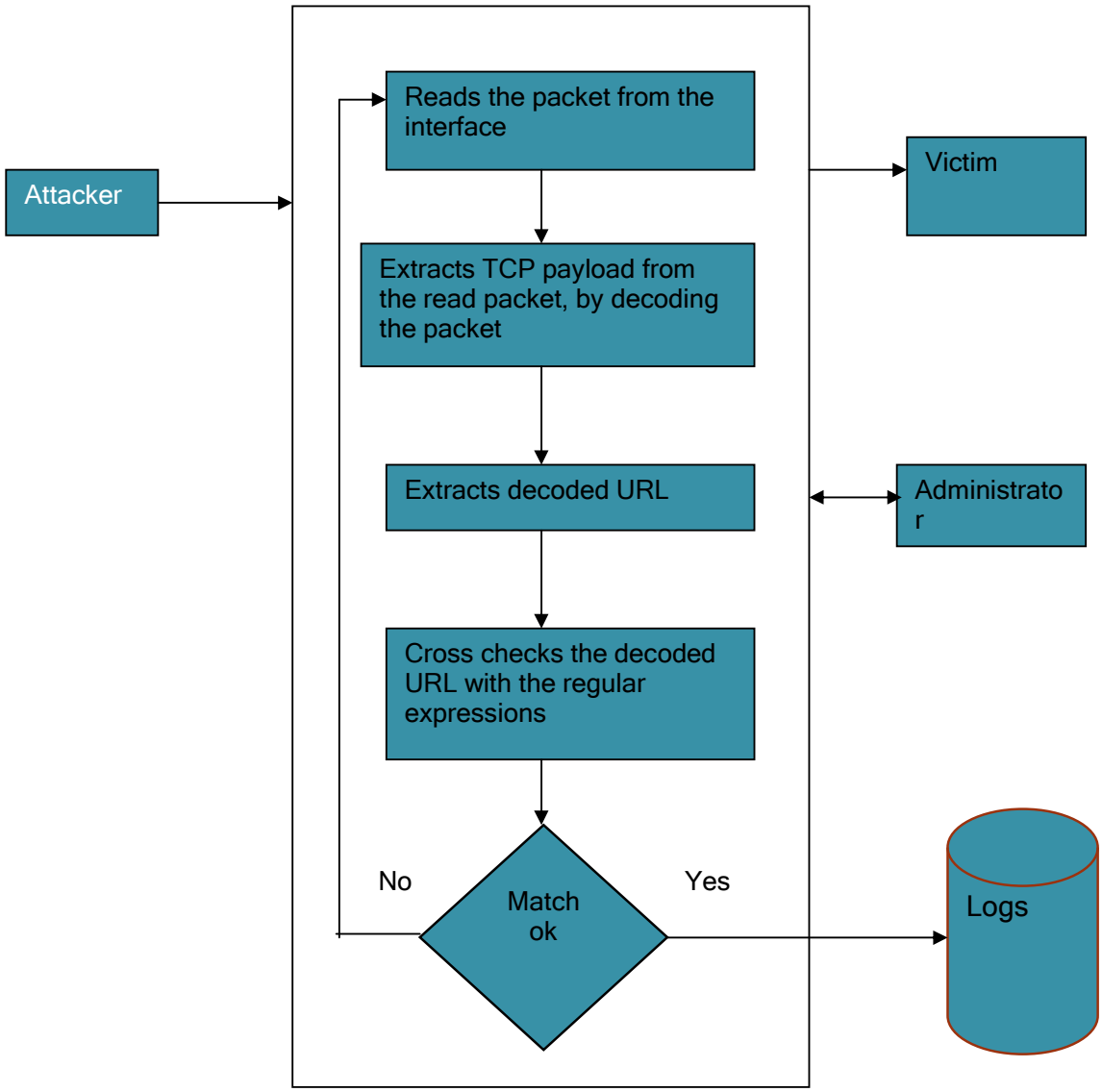



**FIGURE 4**: Detection Process System Design

Decoding the Encoded URLs:
In general URLs will be in the encoded format.

For example
GET http://www.google.com/hl=en&qa=sql%20injection HTTP/1.1

In the above example %20 is an encoded character of the ASCII character space.
Detection System decodes all the captured URL.
Decoded URL: http://www.google.com/hl=en&qa=sql injection

Checking against the regular expressions:
The decoded patterns will be checked against all the regular expressions. If any rule matches, then that log information will be send by the Detection System.

**A.Sample Attack Patterns**
Below attack Patterns are gathered from
http://www.milw0rm.com
Each attack is represented as

<Milw0rmid> ,
<Vulnerable page> ,
<Vulnerable field>,
<Attack pattern>
For example :

1. 7382
tbl_structure.php
table
TABLES%60+where+0+union+select+char%2860%2C+63%2C+112%2C+104%2C+112%2C+32%2C+101%2C+118%2C+97%2C+108%2C+40%2C+36%2C+95%2C+71%2C+69%2C+84%2C+91%2C+101%2C+93%2C+41%2C+63%2C+62%29+into+outfile+%22%2Fvar%2Fwww%2Fbackdoor.php%22+--+1
2. 7378
treplies.asp
message
20814+union+select+1,2,3,4,5,6,7,8+from+msysobjects

**B. Sample Rules to Detect SQL Injection Patterns**

**1**.Detects basic SQL authentication bypass attempts
*Rule*:
(?:^\s*[;>"]\s*(?:union|select|create|rename|truncate|load|alter|delete|updateiinsert|desc))|(?:(?:select|create|rename|truncate|load|alter|delete|update|insert|desc)\s+(?:concat|char|load_file)\s?\(?)|(?:end\s*\);)|("\s+regexp\W)

**2**. Detects conditional SQL injection attempts.
*Rule :*
(?:having\s+[\d\w]\s?=)|(?:if\s?\([\d\w]\s?=)

**3**. Detects basic SQL authentication bypass attempts 1/3
*Rule:*
*(?:^*admin\s*"|(\/*)+"+\s?(?:--|#|\/*|{)?)|(?:"\s*or[\w\s-]+\s*[+<>=(),\s*[\d"])|(?:"\s*[^\w\s]?=\s*")|(?:"\W*[+=]+\W*")|(?:"\s*[!=|][\d\s!=+-]+.*["(].*$)|(?:"\s*[!=|][\d\s!=]+.*\d+$)|(?:"\s*like[+=\s\.-]+[\d"(])|(?:\sis\s*0\W)|(?:where\s[\s\w\.,-]+\s=)

**4**. Detects basic SQL authentication bypass attempts 2/3
*Rule:*
*(?:*union\s*(?:all|distinct)?\s*[([]\s*select)|(?:like\s*"\%)|(?:"\s*like\W*["\d])|(?:"\s*(?:n?and|x?or|not|\|\||\&\&)\s+[\s\w]+=\s*\w+\s*having)|(?:"\s**\s*\w+\W+")|(?:"\s*[^?\w\s=.,;\/)(]+\s*[(@]*\s*\w+\W+\w)|(?:select\s*[\[\]()\s\w\.,-]+from)

**5**. Detects basic SQL authentication bypass attempts 3/3
Rule:
(?:(?:n?and|x?or|not\|\||\&\&)\s+[\s\w+]+(?:regexp\s*\(|sounds\s+like\s*"|[=\d]+x))|("\s*\d\s*(?:--|#))|(?:"[%&<>^=]+\d\s*(=|or))|(?:"\W+[\w+-]+\s*=\s*\d\W+")|(?:"\s*is\s*\d.+"?\w)|(?:"\|?[\w-]{3,}[^\w\s.]+")|(?:"\s*is\s*[\d.]+\s*\W.*")

**6**. Detects concatenated basic SQL injection and SQLLFI attempts
Rule :
(?:^\s*[;>"]\s*(?:union|select|create|rename|truncate|load|alter|delete|update|insert|desc))|(?:(?:select|create|rename|truncate|load|alter|delete|update|insert|desc)\s+(?:concat|char|load_file)\s?\(?)|(?:end\s*\);)|("\s+regexp\W)

**7.** Detects chained SQL injection attempts
*Rule:*
(?:\d\s+group\s+by.+\()|(?:(?:;|#|--)\s*(?:drop|alter))|(?:(?:;|#|--)\s*(?:update|insert)\s*\w{2,})|(?:[^\w]SET\s*@\w+)|(?:(?:n?and|x?or|not|\|\||\&\&)\s+\w+[!=+]+[\s\d]*["=(])

**8.** Detects chained SQL injection attempts 2/2
*Rule:*
(?:*\/from)|(?:\+\s*\d+\s*\+\s*@)|(?:\w"\s*(?:[-
+=|@]+\s*)+[\d(])|(?:coalesce\s*\(|@@\w+\s*[^\w\s])|(?:\W!+"\w)|(?:";\s*(?:if|while|begin))|(?:"[\s\d]+=\
s*\d)

**9.** Detects SQL benchmark and sleep injection attempts including conditional queries
*Rule:*
?:(select|;)\s+(?:benchmark|if|sleep)\s?\(\s?\(?\s?\w+)

## 5. RESULT

In our work the attack patterns are taken from milw0rm website for 45 days duration. Data patterns are stored in a file with date every day. This information is related to commercial and educational websites. Then this data is sent which is attack pattern style to the algorithms. The input given to our system is in form of packets after extracting the url information.

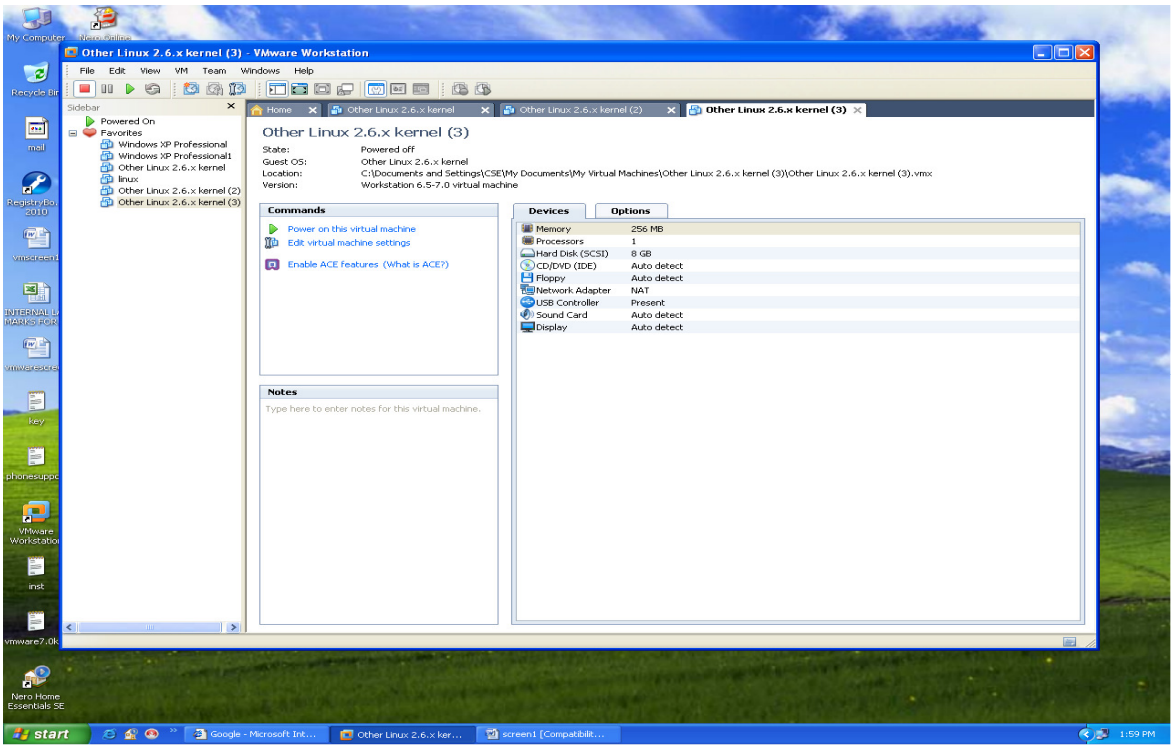



**FIGURE 5**: VMware with Attacker and Detection System

The result of our work is simulating the paths between two hosts and specifying the route paths. If any congestion occurs then it will split the packets and transmits it into multiple paths. Finally our system shows the result with kind of attack patterns which fails to false positives. The work is tested using VMware 7.0 as shown in Figure 5, for cloning the system with linux compatibility and coding is done in C with Perl compatibility. So we have shown our result in the form of text.

It is observed that the rules of Detects advanced XSS probings via Script(), constructors and XML namespaces,,JavaScript location/document property access, basic obfuscated JavaScript script injections, obfuscated JavaScript script injections, JavaScript cookie stealing and redirection attempts, data: URL injections and common URI schemes, possible event handlers, , possibly malicious html elements including some attributes, nullbytes and HTTP response splitting, MySQL comments, conditions and ch(a)r injections, conditional SQL injection attempts, concatenated basic SQL injection and SQLLFI attempts, code injection attempts.

## 6. CONCLUSION AND FUTURE WORK

Here we have developed a highly automated approach for protecting Web applications from SQL injection attacks. This application consists of 1) Using regular expression we found known attacks 2) Allowing only trusted data to form the semantically relevant parts of queries such as SQL keywords and operators. 3) Performs syntax-aware evaluation of a query string immediately before the string is sent to the database for execution. This paper also provides practical advantages over the many existing techniques whose application requires customized and complex runtime environments

To implement the functionality of query fragments that come from external sources, developer must list these sources in a configuration file that SQLIMDS processes before instrument the application. We can enhance SQLIMDS to work on web applications developed using any programming language or framework. The work can be extended by using SQLIMDS to protect actually deployed Web applications, Implementation for binary applications. for high availability on load balancing and disaster recovery. Load balancing can automatically handle failures. (bad disks, failing fans, "oops, unplugged the wrong box", ...), Make that service always work any IP addresses, Anything that has fail over or an alternate server – the IP needs to move (much faster than changing DNS).In Disaster Recovery Planning can have a status update site / weblog, Plans for getting hardware replacements, Plans for getting running temporarily on rented "dedicated servers" (ev1servers, rack space, ...)

## 7.REFRENCES

[1] R.Ezumalai, G. Agila, "Combinational Approach for Preventing SQL Injection Attacks," IEEE 2009-International Advance Computing Conference-2009 .

[2] S.W. Boyd and A.D. Keromytis, "SQLrand: Preventing SQL Injection Attacks," Proc. Second Int'l Conf. Applied Cryptography and Network Security, pp. 292-302, June 2004.

[3] Sagar Joshi, " SQL Injection Attack and Defense ", white paper,2005

[4] Ke Wei, M. Muthuprasanna, Suraj Kothari, " Preventing SQL Injection Attacks in Stored Procedures ", Proceedings of the 2006 Australian Software Engineering Conference (ASWEC'06)

[5] J. Clause, W. Li, and A. Orso, "Dytan: A Generic Dynamic Taint Analysis Framework," Proc. Int'l Symp. Software Testing and Analysis, pp. 196-206, July 2007.

[6] Xiang Fu Xin Lu Boris Peltsverger Shijun Chen , " A Static Analysis Framework For Detecting SQL Injection Vulnerabilities", 31st Annual International Computer Software and Applications Conference(COMPSAC 2007)

[7] "Top Ten Most Critical Web Application Vulnerabilities," OWASP Foundation, http://www.owasp.org/documentation/topten.html, 2005.

[8] V. Haldar, D. Chandra, and M. Franz, "Dynamic Taint Propagation for Java," Proc. 21st Ann. Computer Security Applications Conf., pp. 303-311, Dec. 2005.

[9] W. Halfond, A. Orso, and P. Manolios, "Using Positive Tainting and Syntax-Aware Evaluation to Counter SQL Injection Attacks," Proc. ACM SIGSOFT Symp. Foundations of Software Eng., pp. 175- 185, Nov. 2006.

[10] W.G. Halfond and A. Orso, "AMNESIA: Analysis and Monitoring for NEutralizing SQL-Injection Attacks," Proc. 20th IEEE and ACM Int'l Conf. Automated Software Eng., pp. 174-183, Nov. 2005.

[11] W.G. Halfond, J. Viegas, and A. Orso, "A Classification of SQLInjection Attacks and Countermeasures," Proc. IEEE Int'l Symp. Secure Software Eng., Mar. 2006.

[12] J. Newsome and D. Song, "Dynamic Taint Analysis for Automatic Detection, Analysis, and Signature Generation of Exploits on Commodity Software," Proc. 12th Ann. Network and Distributed System Security Symp., Feb. 2005.

[13] "On the stability of networks operating TCP-like congestion control," in Proc. IFAC World Congress, Barcelona, Spain, 2002.

[14] C. Anley, "Advanced SQL Injection In SQL Server Applications," white paper, Next Generation Security Software, 2002.

[15] Stuart McDonald SQL Injection: Modes of Attack, Defence, and Why It Matters , GIAC Security Essentials Certification (GSEC) Practical Assignment - Version 1.4 (amended April 8, 2002) - Option One

[16] http://nvd.nist.gov

[17] http://www.milw0rm.com

[18] http://www.securityfocus.com

# CALL FOR PAPERS

**Journal:** International Journal of Computer Science and Security (IJCSS)
**Volume:** 4 **Issue:** 5
**ISSN:**1985-1553
**URL:** http://www.cscjournals.org/csc/description.php?JCode=IJCSS

**About IJCSS**

The International Journal of Computer Science and Security (IJCSS) is a refereed online journal which is a forum for publication of current research in computer science and computer security technologies. It considers any material dealing primarily with the technological aspects of computer science and computer security. The journal is targeted to be read by academics, scholars, advanced students, practitioners, and those seeking an update on current experience and future prospects in relation to all aspects computer science in general but specific to computer security themes. Subjects covered include: access control, computer security, cryptography, communications and data security, databases, electronic commerce, multimedia, bioinformatics, signal processing and image processing etc.

To build its International reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJCSS.

**IJCSS List of Topics**

The realm of International Journal of Computer Science and Security (IJCSS) extends, but not limited, to the following:

- Authentication and authorization models
- Communications and data security
- Computer Engineering
- Bioinformatics
- Computer Networks
- Computer graphics
- Cryptography
- Computer security
- Databases
- Data mining
- Image processing
- Electronic commerce
- Operating systems
- Object Orientation
- Programming languages
- Parallel and distributed processing
- Signal processing
- Robotics
- Theory
- Software engineering

## Important Dates

**Volume:** 4
**Issue:** 5
**Paper Submission:** September 30 2010
**Author Notification:** November 01, 2010
**Issue Publication:** November / December

# CALL FOR EDITORS/REVIEWERS

CSC Journals is in process of appointing Editorial Board Members for ***International Journal of Computer Science and Security (IJCSS)***. CSC Journals would like to invite interested candidates to join **IJCSS** network of professionals/researchers for the positions of Editor-in-Chief, Associate Editor-in-Chief, Editorial Board Members and Reviewers.

The invitation encourages interested professionals to contribute into CSC research network by joining as a part of editorial board members and reviewers for scientific peer-reviewed journals. All journals use an online, electronic submission process. The Editor is responsible for the timely and substantive output of the journal, including the solicitation of manuscripts, supervision of the peer review process and the final selection of articles for publication. Responsibilities also include implementing the journal's editorial policies, maintaining high professional standards for published content, ensuring the integrity of the journal, guiding manuscripts through the review process, overseeing revisions, and planning special issues along with the editorial team.

A complete list of journals can be found at http://www.cscjournals.org/csc/byjournal.php. Interested candidates may apply for the following positions through http://www.cscjournals.org/csc/login.php.

*Please remember that it is through the effort of volunteers such as yourself that CSC Journals continues to grow and flourish. Your help with reviewing the issues written by prospective authors would be very much appreciated.*

Feel free to contact us at coordinator@cscjournals.org if you have any queries.

# Contact Information

**Computer Science Journals Sdn BhD**
M-3-19, Plaza Damas Sri Hartamas
50480, Kuala Lumpur MALAYSIA

Phone: +603 6207 1607
+603 2782 6991
Fax: +603 6207 1697

**BRANCH OFFICE 1**
Suite 5.04 Level 5, 365 Little Collins Street,
MELBOURNE 3000, Victoria, AUSTRALIA

Fax: +613 8677 1132

**BRANCH OFFICE 2**
Office no. 8, Saad Arcad, DHA Main Bulevard
Lahore, PAKISTAN

**EMAIL SUPPORT**
Head CSC Press: coordinator@cscjournals.org
CSC Press: cscpress@cscjournals.org
Info: info@cscjournals.org

COMPUTER SCIENCE JOURNALS SDN BHD
M-3-19, PLAZA DAMAS
SRI HARTAMAS
50480, KUALA LUMPUR
MALAYSIA