Assessment and Mitigation of Risks Involved in Electronics Payment Systems

Tanvir Tahir[1], Kamar Abbas[2]

[1]Department of Computer Science & Engineering, Institute of Engineering & Technology, Lucknow, UP, India
[2]Azad Institute of Engineering & Technology, Lucknow, UP, India

Abstract: ***This paper deals with the risk assessment of different types of electronics and mobile payment systems as well as the countermeasures to mitigate the identified risk in various electronics and mobile payment synthesis.***

Keywords: e-commerce, CC Redbook, Mitigation Risk, OSSTMM, NIST

1. Introduction

Over the last years, a criminal revolution has taken place in cyberspace. Online criminals now employ basic economic concepts to develop their fraudulent businesses. Malicious hackers rely on techniques such as *specialization* of goods and services. Examples includes Phishermen who create fake websites and Botnet Herders who manage large collections of compromised computers; *outsourcing of production*, exemplified by the growth of automated crime-ware tools; *multivariate pricing*, best illustrated by credit card fraud schemes, where factors such as issuing bank, geographic location, and the credit card's rareness determine its value; and *bulk pricing*, demonstrated through offerings of large collections of e-mail addresses and credit card numbers to discount prices.

When it comes to non-trivial networked computer systems, bulletproof security is very hard to achieve. Over a system's lifetime new security risks are likely to emerge from newly discovered classes of vulnerabilities or the arrival of new threat agents. Given the dynamic environment in which computer systems are deployed, continuous evaluations and adjustments are wiser than one-shot efforts for perfection. Security risk management focuses on assessing and treating security risks against computer systems. In this paper, elements from risk management are applied to two real-world systems to;

- Identify,
- Evaluate, and
- Mitigate risks.

One of the pinpointed weaknesses is studied in-depth to produce an exploit against the affected system. In addition, approaches to handle common software security problems are described. How can appropriate security features, attributes, safeguards, and countermeasures be "designed in" to an e-commerce system? To answer this question we have to examine the existing/current standard and security practices.

2. Current Security Standards and Practices; An Overview

Electronic commerce (EC) is about doing business electronically. It is based on the ability to electronically process and transmit data, including text, sound and video. EC encompasses many diverse activities including electronic trading of goods and services, online delivery of digital content, electronic fund transfers, auctions, direct consumer marketing, and after-sales service. It involves products (consumer goods) and services (information services), traditional activities (healthcare, education) and new activities (virtual malls).

An example of generic e-commerce system architecture is a client communicating with a merchant server via the internet. The merchant server (also known as business server), usually utilizes a web server to accept client requests, a database management system to manage data and a payment gateway to provide online payment services.

For many years companies have exchanged business data over a variety of communication networks. Now, there is an accelerated expansion driven by the exponential growth of the internet. EC is now rapidly expanding into a complex web of commercial activities transacted on a global scale between an ever-increasing number of participants, corporate and individual, known and unknown, on global open networks such as the Internet [1].

This explosion requires a comparable growth in security presence. Yet, it is likely that readers of this paper have heard about successful e-commerce break-ins at some time in the past few years. Such events have made the headlines of newspapers, magazines and books.

Until 2001, there was no published security design standard encompassing all security features. Best practices developed as new security issues were encountered and overcome.

However, improvements to security design strategies did not come from a centralized source such as the NIST [2] security design standard. This section gives an overview of current security standards. There are currently three security standards for IT systems that address overall system security:

- The Common Criteria(CC) Redbook [3],

- The NIST "Underlying Technical Models for Information Technology Security" document [2] and
- The Open Source Security Testing Methodology Manual (OSSTMM) [4]

2.1. Common Criteria Redbook

Many consumers of IT lack the knowledge, expertise or resources necessary to judge whether their confidence in the security of their IT products or systems is appropriate, and they may not wish to rely solely on the assertions of the developers. Consumers may therefore choose to increase their confidence in the security measures of an IT product or system by ordering a trusted third-party analysis of its security (i.e. a security evaluation).

The Common Criteria (CC) Redbook was one of the first attempts to standardize security assessment / evaluation requirements for Information Technology (IT) systems. It can be used to select the appropriate IT security measures and contains criteria for evaluation of security requirements. It consists of the following three parts:

Part-I: *Introduction and general model* is the introduction to the CC. It defines general concepts and principles of IT security evaluation and presents a general model of evaluation. Part-I also presents constructs for expressing IT security objectives, for selecting and defining IT security requirements, and for writing high-level specifications for products and systems. In addition, the usefulness of each part of CC is described in terms of each of the target audiences.

Part –II: *Security functional requirements*, establishes a set of security functional components as a standard way of expressing the security functional requirements for Targets of Evaluation (TOEs) or systems.

Part-III: *Security assurance requirements*, establishes a set of assurance components as a standard way of expressing the assurance requirements for TOEs. Part-III also defines evaluation criteria for Protection Profiles (PPs) and Security Targets (STs) and presents evaluation assurance levels that define the predefined CC scale for rating assurance for TOEs, which is called the Evaluation Assurance Levels (EALs).

However, in the domain of e-commerce systems, the CC Redbook is intended to be used to evaluate system security only after the system is implemented. It cannot be used directly during the system design phase. This means that CC does not address the core problem of the lack of a security design-time standard for ecommerce systems.

2.2. NIST Underlying Technical Models for IT Security

The purpose of the NIST "Underlying Technical Models for Information Technology Security" document is to provide a description of the technical foundations, termed 'models', which underlie secure information technology (IT). The intent is to provide, in a concise form, the models that should be considered in the design and development of technical security capabilities. These models encompass lessons learned, good practices, and specific technical considerations. The security goal according to NIST can be met by achieving five security objectives:

i. **Availability** of systems and data for intended use only. This is a requirement to assure that systems work promptly and service is not denied to authorize users and denied for unauthorized / malicious users.
ii. Integrity of system and data. This objective has two facets:
   - Data integrity: the property that data has not been altered in an unauthorized manner while in storage, during processing, or while in transit.
   - System integrity: the quality that a system is performing the intended function in an unimpaired manner, free from unauthorized manipulation.
iii. Confidentiality of data and system information. This is the requirement that confidential information not be disclosed to unauthorized individuals. Confidentiality protection applies to data in storage, during processing, and while in transit.
iv. Accountability to the individual level. This is the requirement that actions of an entity can be traced uniquely to that entity.
v. Assurance that the other four objectives have been adequately met. This objective is the basis for confidence that the security measures, both technical and operational, work as intended to protect the system and the information it processes.

The other four security objectives are adequately met by a specific implementation when:

a) There is correct implementation of the required functionality
b) There is sufficient protection against unintentional errors
c) There is sufficient resistance to intentional penetration or by-pass

Assurance is a continuum; the amount of assurance needed varies between systems.

2.3. NIST Security Services Model

The NIST security services model shows the primary services and supporting elements used in implementing an information technology security capability, along with their primary relationships. The model also classifies the services according to their primary purpose as follows:

- **Prevent-** These services focus on preventing a security breach from occurring
- **Recover-** The services in this category focus on the detection and recovery from a security breach.
- **Support-** These services are generic and underlie most information technology security capabilities.

Prevention Services

Prevention services are intended to prevent a security breach from ever happening.

- ***Protected communications:*** In a distributed system, the ability to accomplish security objectives is highly dependent on trustworthy communications. The *protected communications service* ensures the *integrity*, *availability*, and *confidentiality* of information while in transit. In most situations all three elements are essential requirements, with *confidentiality* being needed for authentication information.
- ***Authentication****:* Ensuring that a claimed identity and is valid is extremely important. The *authentication service* provides the means to verify the identity of a subject.
- ***Authorization****:* The *authorization service* enables specification and subsequent management of the allowed user or process actions for a given system.
- ***Access control enforcement:*** When the subject requesting access has been validated for access to particular processes, enforcing the defined security policy is still necessary. Access control enforcement provides this enforcement. Frequently, the enforcement mechanisms are distributed throughout the system. It is not only the correctness of the access control decision, but also the strength of the access control enforcement mechanism that determines the level of security achieved.
- ***Non-repudiation****:* System accountability depends upon the ability to ensure that senders cannot deny sending information and that receivers cannot deny receiving it. *Non-repudiation* is a service that provides prevention and detection. This service has been placed into the prevention category because the mechanisms implemented prevent the ability to successfully repudiate an action. As a result, this service is typically performed at the point of transmission or reception.
- ***Transaction privacy****:* Both government and private systems are increasingly required to maintain the privacy of individuals using these systems. The *transaction privacy service* protects against loss of privacy of an individual.

2.4. Detection and Recovery Services

Because no set of prevention measures is perfect, it is necessary to both *detect* security breaches and to take actions to *recover from* and *reduce* their impact.

- ***Audit****:* Auditing security-relevant events is a key element for detection of and recovery from security breaches.
- ***Intrusion detection and containment****:* Detecting insecure situations is essential in order to respond in a timely manner. Also, detecting a security breach is of little use if no effective response can be initiated. The intrusion detection and containment service provides these two capabilities.
- ***Proof of Wholeness:*** In order to determine that integrity has been compromised, the ability must exist to detect when information or system state is potentially corrupted. The proof of wholeness service provides this ability.
- ***Restorability to a 'secure' state:*** When a security breach occurs, the system must be able to return to a state that is known to be secure. That is the purpose of this service.

2.4.1 Supporting Services

Supporting services are pervasive and inter-related with many other services. The NIST supporting services are:

- ***Identification (and naming):*** It is essential that both subjects and objects be uniquely identifiable in order to implement many of the other services. This service provides the capability of uniquely identifying users, processes, and information resources.
- ***Cryptographic key management:*** When cryptographic functions are implemented in various other services, cryptographic keys must be securely managed.
- ***Security administration:*** In order to meet the needs of a specific installation and to account for changes in the operational environment, security features of the system need to be administered. As an example, a password database might make use of the operating system platform API in one installation and a SQL database in another installation.
- ***System protections:*** Underlying the various security functional capabilities is a base of confidence in the technical implementation. This represents the quality of the implementation from both the perspective of the design processes used and the manner in which the implementation was accomplished.

Some examples of *system protections* are: residual information protection (also known as object reuse), least privilege, layering, process separation, modularity, and minimization of what needs to be trusted (also known as least-trust).

2.5. Open Source Security Testing Methodology Manual (OSSTMM)

The Open Source Security Testing Methodology Manual (OSSTMM) attempts to set a standard for internet security testing on a running system. Although this type of testing cannot be done during the system design phase, the OSSTMM itself will be used as a reference for security attacks. An important aspect of the methodology is the way interdependencies are mapped and related to each other.

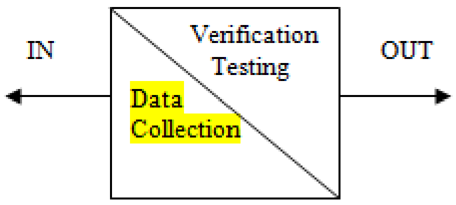

Figure 1: A Flow Representation of Modules in OSSTMM

A ***security test*** is performed with two types of *attacks* according to OSSTMM. A ***passive attack*** is often a form of data collection that does not directly influence the target. An ***intrusive attack*** however trespasses upon the target and can be monitored, logged, and used to alarm the target. [Herzog, 2001] The process (or methodology) of a security test concentrates on evaluating areas that directly affect security presence. These areas are of following types:

- ***Visibility:*** Visibility is what can be seen, logged, or monitored in the security presence both with and without the aid of electronic devices. This includes, but is not limited to, communication devices (such as e-mail) and network packets (such as TCP/IP).
- ***Access:*** Access is an entry point into the security presence. This can include, but is not limited to, a web page, a network connection, or anything where a computer interacts with another computer within a network.
- ***Trust:*** Trust is a specialized attribute with respect to security presence as a whole. Trust includes the kind and amount of authentication, non-repudiation, access control, accountability, confidentiality, and integrity between two or more features within the security presence.
- ***Alarm:*** Alarm is the timely and appropriate notification of activities that violate or attempt to violate visibility, access, or trust. In many security breaches, alarm is often the single process which initiates further consequences.

The methodology starts with a *security map* of the system and is broken down into *sections*, *modules* and *tasks*. The *sections* are specific points in the *security map* which overlap with each other. The *modules* represent the *flow* of the methodology from one *security presence point* to the other. Each module has an *input* and an *output*. The *input* is the information used in performing each task. The *output* is the result of completed tasks [4].

Output may or may not be analyzed data (also known as intelligence) to serve as an input for another module. It may even be the case that the same output serves as the input for more than one module or section. Some tasks yield no output; this means that modules may exist for which there is no input. Modules which have no input can be ignored during testing.

Ignored modules do not necessarily indicate an inferior test; rather they may indicate superior security [4]. This is because the input of one module depends on the output of another. Having no output from a certain module can either be traced back to an inferior test in the case of weak security testing, or to superior security in the case of having proper security countermeasures.

The methodology flows from the initial module to the completion of the final module. It allows for a separation between data collection and verification testing of and on the collected data. Each module has a relationship to the one before it and the one after it. Each section has inter-relational aspects to other modules and some may inter-relate with all the other sections [4]. Overall, security testing begins with an input that is ultimately the addresses of the systems to be tested. Security testing ends with the beginning of the analysis phase and the final report.

As discussed in this section, OSSTMM applies to the ethical hacking process of a running internet system. Yet, this can only be done after the system is implemented. OSSTMM is different from the Common Criteria document and the NIST "underlying technical models for IT security" document in the fact that it emphasizes blocking malicious user requirements. Still, it cannot be used directly during the system design phase and, thus, does not solve our core problem, namely the need for methods and tools based on a security design standard in e-commerce systems.

3. Countermeasures to Mitigate the Identified Risk in Various Electronic and Mobile Payment Systems

3.1. Secure Computer Systems

The production of secure networked computer systems is a difficult undertaking. A striking difference from traditional engineering disciplines is the presence of skilled and creative attackers, who relentlessly try to break systems. Other factors that add to the challenge of creating secure computer systems include the trinity of trouble:

- The growing *connectivity* of computers and software. The ongoing push to publish systems on the internet results in increased risks, as adversaries find it easier to launch attacks;
- The increasing degree of program *extensibility* results in flexible software that can swiftly accommodate new business needs, but also open up for unwanted malicious extensions; and
- The rising *complexity* of computer systems. More lines of code translate to a higher probability of introducing vulnerabilities.

Systems fail for a number of reasons. Anderson [5] argues that cryptosystems fail more often because of implementation faults and managerial issues than because of insufficient cryptographic primitives. In the case of software security problems, recent numbers show a 50/50 split between implementation bugs and design flaws. A bug is an implementation-level programming mistake that most often can be easily fixed. Buffer overflows are bugs that can usually be removed by introducing proper array bounds checking. A flaw is a weakness introduced at the design-level that cannot normally be fixed through simple adjustments to the program code. A system redesign is often needed to remove flaws. Whereas many bugs can be identified by automated tools, finding flaws typically require manual inspection by trained professionals.

It is widely recognized that the cost of fixing defects increases with each new stage in the Software Development Life Cycle (SDLC). Studies show that a flaw left unfixed in the design phase can become more than ten times as expensive to remove during system maintenance. This suggests that companies are wise to have a strong focus on detecting and resolving security problems as early as possible in the development process. As a consequence, identifying design flaws is more attractive from a cost-saving perspective than discovering implementation bugs.

3.2. Security Risk Management

Risk management of system architecture and design has shown great promise in finding and removing flaws. At Microsoft, this technique has been identified as a critical success factor for the company's strides in software security. Risk management involves assessment and treatment. In the risk assessment phase, the risk management team creates an overview of the system, identifies system threats & vulnerabilities, and evaluates & ranks threat/vulnerability pairs. The risk treatment phase outlines steps to mitigate unacceptable risks.

3.3. Computer Security

Computer system is the collection of hardware, software, information, and people that an organization make use of to carry out computing tasks.

Computer security concerns the building of computer systems that continue to function under malice, error, or mischance. Combined, these definitions hint at the large challenges involved in making secure systems. Firstly, the broad scope significantly adds to the problem of creating secure computer systems. Making secure software is a hard problem in itself, but does not solve the computer security problem alone. Phishing a scam that tricks end users into revealing their secret credentials is a form of attack that deceives the human component of a system, and cannot be dealt with exclusively in software. Secondly, a thorough treatment of computer security requires expertise from many disciplines. Examples include lawyers who can help to ensure that legal requirements are understood and met; economists who can aid in analyzing and shifting the financial incentives for potential attackers; and cryptographers who can assist in safe use of cryptographic primitives.

In “The Protection of Information in Computer Systems”, Salzer and Schroeder describe security as techniques that control who can use or modify information in a computer system. In particular, categories for security violations include unauthorized:

• Information release,
• Information modification, and
• Denial of use.

The authors go on to argue that protecting a computer system against security violations is a difficult undertaking, and that no complete method exists that allows the construction of secure large general-purpose systems. Instead, they point to eight design principles to guide developers in creating more secure systems. Examples include keeping the design as simple as possible, a warning against believing that holding the design secret will make a system secure, and a recommendation in favor of intuitive and easy to use interfaces.

Today, over 30 years later, we still rely on best practices when developing all but the simplest systems. Introductory textbooks in computer security all describe the three cornerstones of security: confidentiality, integrity, and availability that cover the same ground as the three previously described security violations.

These are not independent concepts, as they sometimes exclude one another, and therefore must be carefully balanced. As an example, encrypted information will only be available to those who possess the secret(s) needed to decrypt that data. A number of security services have been developed to realize and extend the three fundamental goals of security.

In particular, these concepts are central in this paper:

- Authentication is the process of establishing an understood level of confidence in the truth of some claim; and
- Non-repudiation provides protection against someone that falsely denies that a communication took place.

3.4. Risk Management

Risk can be viewed simply as the possibility of suffering harm or loss. This type of risk is often referred to as a pure or non-speculative risk. Examples include uncontrollable events such as natural catastrophes. A speculative risk on the other hand involves a conscious choice that could lead either to a gain or a loss, e.g. investing in stocks.

This paper deals solely with risks from which only a loss can occur. Proper management of non-speculative risks can control and minimize the negative effects of unwanted events.

According to NIST, risk management is the process of identifying risks, assessing risks, and taking steps to reduce risks to an acceptable level. At a bird’s-eye view, risk management can be divided into risk assessment and risk treatment.

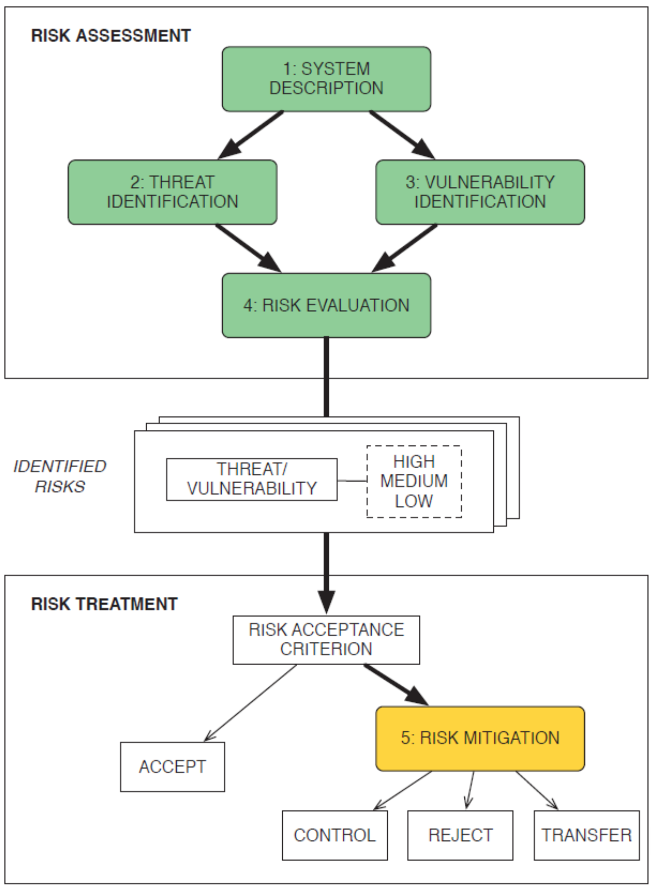

Figure 2: The Risk Management Process

3.4.1. Risk Assessment

The risk assessment phase involves

- **System description**: This first step sets the scope for the risk management process.
- **System information and assets**: such as hardware, software, network topology, and users of the system are identified.
- **Threat identification**: Aims to single out entities who may (accidentally or intentionally) exploit vulnerability in the system. Examples include organized crime, script kiddies, and insiders.
- **Vulnerability identification:** The task of creating a list of system flaws and weaknesses that could allow threats to break the system's security.
- **Risk evaluation**: Based on the likelihood of occurrence and the resulting impact, each threat/vulnerability pair is assigned a risk level.

3.4.2. Risk Treatment

Risk treatment concerns finding ways to deal with the identified threat/vulnerability pairs and their associated risk levels. A risk acceptance criterion signifies how much risk someone is willing to tolerate. If a given threat/vulnerability pair carries a risk below this threshold, it is left untreated and accepted as is.

- Risks that do not fall into the accept category can be handled in three ways:
- Control the risks by introducing countermeasures to reduce the risks down to a tolerable level;
- Reject the risks and use workarounds to avoid the identified problems; or
- Transfer the risks to other parties through for example insurance.

3.5. The Risk Management Process

Figure 2 shows the steps and activities involved in the risk management process. Note that steps 2 and 3 can be executed in parallel. However, they are closely related and must be paired in the risk evaluation phase. A vulnerability for which there are no threats that can exploit the given weakness, does not constitute a risk against a system, and should therefore be excluded. An identified threat with no matching vulnerability should be treated in the same manner.

Risk evaluations are based on the likelihood of occurrence and the negative impact resulting from a threat exercising a given vulnerability. Risks can be assessed quantitatively or qualitatively. A quantitative approach relies on mathematics to exactly express risks.

Threat/vulnerability pairs are assigned probabilities of occurrence and impact values. An example is annual loss expectancies that can be derived by multiplying a potential monetary loss with the probability of occurrence during a year. It can be very difficult to quantify risks. An example is the challenge of accurately estimating loss of reputation.

A qualitative approach describes risks using a hierarchical scale. As an example, likelihood of occurrence can be approximately described through three categories: low, medium, and high. A similar approach can be applied to impact estimation.

Figure 3 shows a risk matrix based on the example three categories for likelihood and impact. The illustration demonstrates a possible five-level risk matrix derived from the various combinations of likelihood and impact. In order for the risk evaluation results to be reproducible, the different risk levels must be defined. The assessments in this thesis follow the qualitative approach. In light of the choice of qualitative risk analysis, an appropriate definition is to define risk as the negative impact of the exercise of vulnerability, considering both the likelihood and the impact of occurrence.

High	Medium	High	Very High
Medium	Low	Medium	High
Low	Very Low	Low	Medium
	Low	Medium	High

Figure 3: Five Level Risk Matrix Impact

As illustrated in Figure. 2, the results of the risk evaluation phase are a set of threat/ vulnerability pairs with an associated risk level. The identified risks can then be ranked and compared to the risk acceptance criterion. Different methods exist for selecting this criterion. One approach is to set a limit against which all risks are contrasted. Those ranked above the limit are subject to risk mitigation techniques. The *As Low As Reasonably Practicable* (ALARP) principle requires all risks to be mitigated to the point where the costs of applying further risk treatment grossly outweighs the expected benefits. This latter form of criterion has been successfully applied in the oil industry. Systems in that industry involve risks with low likelihood and very high impact that should be avoided.

3.6. Residual Risks

It is important to note that all risks cannot be removed in a non-trivial system. Plans should be devised to handle these residual risks. An important task in this regard is to assign ownership of the remaining risks. Some system owners assume this responsibility themselves. An example is credit card companies that cover customers' losses from card misuse, as long as the bureaus themselves cannot trace the reported misuse back to the affected client. Another approach is to split the cost between the system owner and users, as widely adopted by the insurance industry in the form of individual shares. Yet another alternative is to communicate the remaining risks to customers, and let them bear the costs when losses occur. This approach is frequently used in amusement parks, where customers get risk warnings prior to entering rides.

The numbers on costs of fixing software defects, mentioned in the introduction of this paper, indicate that risk management is most effective early in the SDLC. However, it should be revisited and updated at later stages of development. Failure to do so may lead to uninformed decision making, as new threats and/or vulnerabilities surface, rendering previous risk analyses inaccurate or obsolete. A good risk management regime therefore plans for changes and continuously updates its evaluation of risks as threats and vulnerabilities evolve.

4. Conclusions

It has been observed that following steps must be taken into considerations for risk assessment and mitigation involved in electronic payment systems:

a. System description
b. Threat identifications
c. Vulnerability identification
d. Risk evaluation
e. Identification of level of risk (threat/vulnerability) how low, how medium and how high
f. Risk treatment applied
g. Risk acceptance criteria
h. Risk mitigation (control, reject & transfer)

5. Acknowledgement

The authors are thankful to Institute of Engineering & Technology, Sitapur Road, Lucknow, UP and Integral University, Lucknow, UP for providing laboratory and other facilities.

References

[1] European Commission, “Information Society Statistics”, website:http://europa.eu.int/information_society/ topics/ebusiness/ecommerce/3informati n/statistics/index_en.htm.
[2] National Institute of Standards and Technology (NIST), Special Publication 800-33,“Underlying Technical Models for Information Technology Security”, 2001. website:http://csrc.nist.gov/publications/nistpubs/800-33/sp800-33.pdf.
[3] Common Criteria (CC), “Common Criteria for Information Technology Security Evaluation”, August 1999, website: http://www.commoncriteria.org/.
[4] P. Herzog, “The Open Source Security Testing Methodology Manual”, version 1.5, May 2001, website: http://ideahamster.org/.
[5] Anderson, M.M. (1998), “*Electronic Cheque Architecture, Version 1.0.2”*, Financial Services Technology Consortium.