# Authentication and Authorization:Tool for E-Commerce Security

Pradnya B. Rane
Computer department
VJTI
Matunga, India

Pallavi Kulkarni
Computer department
VJTI
Matunga, India

Suchita Patil
Computer department
VJTI
Matunga, India

Dr. B.B.Meshram
Computer department
VJTI
Matunga,India

***Abstract*—Web-based e-commerce and distributed applications are changing the way we buy goods, access information and learn. The general purpose of electronic commerce or, as everybody knows it e-commerce, is to provide a way of interconnecting the two major components of a market economy: the demand and the supply, without physically putting together both of the ends of the business process .Today's distributed e-commerce applications typically rely upon various technologies in their realization, including the web, scripting languages, server-side processing and an underlying database. The combination of these technologies creates a system that requires attention to the security issues of each component and the system as a whole. Hence we can use authentication and authorization tool for security of E-commerce application.**

***Keywords-component; Authentication, Authorization,Sniffing, ID Spoofing, Brute Force Attack***

## *I.* INTRODUCTION

E-commerce lets businesses reduce costs, attain greater market reach, and develop closer partner relationships. However, using the Internet as the underlying backbone network has led to new risks and concerns. Too often E-commerce security is framed solely as a communications security problem. Cryptography is seen as the essential security technology. Encryption algorithms and digital signatures provide the basic building blocks, protocols like SSL constitute the next layer of mechanisms that in turn support applications like secure E-mail e .g. SIMIME, electronic payment schemes like SET (secure electronic transfer), a protocol for payment-card transactions developed by Visa and Mastercard. The final ingredients are public key infrastructures (PKIs) that tic cryptographic keys to user. From this point of view, current export restrictions on equipment and software implementing cryptographic algorithms are the obstacles that has to be removed before universally deployed strong cryptography facilitates secure E-commerce. Often, industry analysts cite trust and security as the main hurdles in growing e-commerce[1][4][5].

This paper is organized as follows. Section I is introduction which gives brief ides about E-commerce applications. Section II focused on security challenges in the E-commerce Applications. Also it focuses on attacks related to authentication and authorization, their attack enablers and their countermeasures using security-oriented authentication and authorization design model for e-commerce.. Section III discusses the security areas and limitations of e-commerce application. Paper concludes in section IV.

## II. RELATED WORK

Use of email and other related technology increasingly facilitates collaboration and is more commonly being used for official communications. Official communications via the internet are too often done in insecure mode. Web-based e-commerce applications commonly employ multiple tiers (3-tier client server architecture) and a combination of technologies such as HTML, XML, JavaScript, Java (JSP, Servlets), ASP, dynamic html, CGI, and relational databases, as shown in Figure 1. Each of these technologies have separate and in some cases incompatible approaches to protection against intrusion[2].

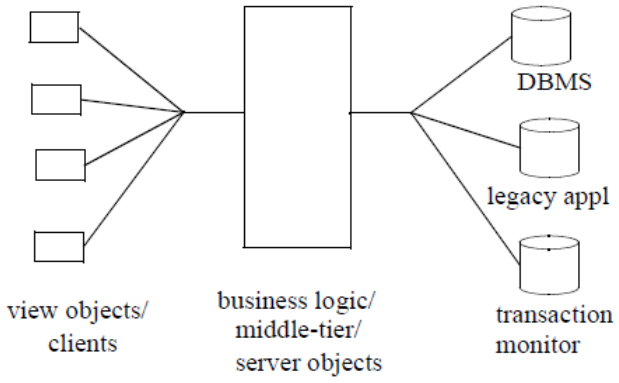


Figure1: 3-tier client-server architecture[2].

For web-based applications, the communication between clients and the middle-tier is via web protocol http. Clients

may employ any number of technologies such as applets, html, xml, and scripts. The middle-tier business logic often employs any of a number of CGI work-arounds such as Netscape's NSAPI, Microsoft's ISAPI, WebObjects, ASPs, Java J2EE, servlets and JSP. The combination of different technologies at each tier, presents special challenges to security of the overall application. Now we will discuss attacks related to authentication and authorization with their attack enablers and countermeasures used in e-commerce application .

*A. Security- Oriented Authentication Design Model*

Authentication is the process of verifying the identity of a ser, process, or device, often as a prerequisite to allowing access to resources in a system. The identity of a certain user or process is challenged by the system and proper steps must be taken to prove the claimed identity. Authentication models may depend on specific technologies. The specific security attacks related to authentication in e-commerce systems are as follows[6][9][10].

- Sniffing attacks (also known as man-in-the-middle attacks)
- Dictionary attacks
- Replay attacks
- Brute-force attacks
- ID spoofing attacks (also known as spoofing attacks)
- Credential decryption attacks (supplementary to other types of attacks)
- Side-channel attacks

For each authentication attack, derive its enablers and countermeasures This section provides a succinct abstract description of all known authentication-related security attacks. Attack enablers are then identified, and effective countermeasures are prescribed.

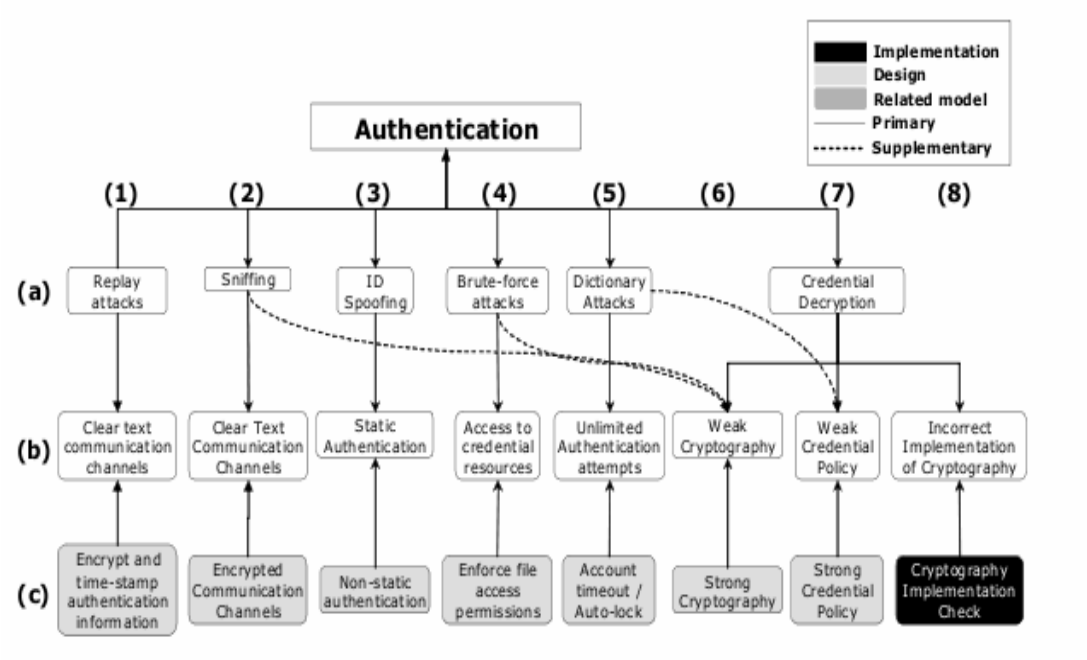


Figure 2: Organizational chart for authentication security attacks (a),attack enablers (b), and countermeasures (c)[6].

Figure 2. shows all security attacks related to authentication in e-commerce systems (a) along with the attack enablers (b) and prescribed countermeasures (c).

*1) Sniffing Attack.*

- Sniffing attacks (also known as the man-in-the-middle attacks) are the digital analogues to phone tapping or eavesdropping. This attack captures information as it flows between a client and a server. Usually, a malicious user attempts to capture TCP/IP transmissions, because they may contain information such as usernames, passwords, or the actual contents of an e- mail message. A sniffing attack (a2) is often classified as a man-in-the-middle attack because in order to capture packets from a user, the machine capturing packets must lie in between the two systems that are communicating (a man- in-the-middle attack can also be waged on either one of the two systems).

- The attack enabler in this case is the process of sending data across communication channels in clear text format (b2).
- However, by encrypting the communication channel (c2) between the user/process and the system,

sniffing attacks are disabled, i.e., sniffing retrieves only useless encrypted information. However, the information can be duplicated and substituted for subsequent transmissions.

*2) ID Spoofing Attacks*

- ID spoofing attacks occur when a malicious user or process claims to be a different user or process (a3). This attack allows an intruder on the Internet to effectively impersonate a local system's IP address. If other local systems perform session authentication based on the IP address of a connection (e.g. rlogin with .rhosts or /etc/hosts.equiv files under Unix), they will believe incoming connections from the intruder actually originate from a local "trusted host" and will not require a password.
- The attack enabler for this attack is for authentication to rely on static information (b3) such as IP addresses, host names, etc. This is equivalent to trusting certain hosts or processes according to some pre-defined static information. The system authenticates the user or process only by checking the given static information. In such a case, the attacker will attempt, through complex attack tools, to "spoof" the system by claiming that he/she is the trusted host or process. Since no challenge is attempted in this case, the attack has a great chance of succeeding.
- The countermeasure for such an attack is to use challenge-based authentication (c3). Challenge-based authentication includes the use of certificates, user/password combinations, etc. If challenge-based authentication is inapplicable for a certain specific case, then least privilege static authentication must be applied. Least privilege static authentication means giving the least possible access privilege to the fewest possible number of users, processes or hosts after successful authentication.

*3) Brute-Force Attacks*

- A Brute-force attack is any form of attack against a password file that attempts to find a valid username and password by successive guessing (a4).
- This type of attack is enabled by gaining access to the credential (user names and passwords) storage medium (b4). The attacker first retrieves a copy of the database system or system file holding credential information. If the credential information is encrypted, a brute -force attack tool will try all possible combinations of user names and passwords. For each combination, the user name and password are encrypted using the same encryption algorithm that was used to encrypt the original credential information. Then, the encrypted data is compared to the retrieved copy of credential data. Different types of encryption algorithms are used and the attack proceeds until both credentials (user name and password) match.
- The countermeasure for this type of attack is to enforce access permissions through a strong access control policy at the operating system level (c4). By doing so, malicious users will fail to retrieve a copy of credential information and, thus, the brute-force attack is disabled.

*4) Dictionary Attacks*

- A dictionary attack is the "smart" version of brute-force attacks and is directed towards finding passwords in a specific list, such as an English dictionary. Dictionary attacks (a5) are also executed using automated tools. Moreover, these tools are capable of working on web interfaces without access to the encrypted format of credential information. These tools require the prior knowledge of the user name only. Once given a user name, the attack tool will try all possible combinations of that user name with a huge database (such as a dictionary) of possible passwords. This attack has a high probability of succeeding since we, as humans, tend to use passwords that are easy to remember.
- The attack enabler is a "high" number of allowed consecutive unsuccessful authentication attempts (b5).
- The countermeasure, in this case, is to prevent the automation of the attack by setting an upper limit on the allowed number of successive unsuccessful authentication attempts. This can be done through an account auto-lock or a timeout procedure (c5). In other words, when a certain number of consecutive, unsuccessful authentication attempts is reached, the system will automatically lock or disable the account and will alarm the system administrator. This will prevent the dictionary attack from proceeding and, thus, the attack is disabled. Enabling or unlocking the account can be done either by the user or automatically by the system after a certain period of time.

*5) Replay Attacks*

- A replay attack (a1) occurs when a malicious user captures an authentication sequence that was transmitted through the network by an authorized user, and then replays the same sequence to the server to get himself/herself authenticate.
- The attack enabler in this case is, again, access to the communication channel and data sent in clear text format (b1).
- The proper countermeasure is to encrypt and time-stamp all sensitive data sent across the communication channel (c1). By doing this,

"replayed" messages can be recognized and discarded and this type of attack is disabled.

*6) Credential Decryption Attacks*

- Credential decryption is a basic supplementary attack for sniffing attacks, brute-force attacks, and dictionary attacks (a7). A tool whose aim is to break the encryption algorithm that was used to encrypt credential information usually performs these attacks.
- Attack enablers for this attack might be a weak cryptographic algorithm (b6), a weak credential policy (b7), or an incorrect implementation of the cryptographic algorithm b8). Weak cryptography increases the probability of success for a brute -force attack or a sniffing attack by allowing the use of cryptographic systems that are easy to crack.
- Its countermeasure is to use a strong cryptographic algorithm that is hard to crack (c6). Please note that a weak cryptography is not the same as a weak credential policy. A weak credential policy, on the other hand, increases the probability of a dictionary attack's success by allowing the existence of easy-to-guess passwords. Its countermeasure is to have a strong credential policy (c7) that forces legitimate system users to create and maintain a safe password that is easy to remember for a legitimate user and difficult to guess for a malicious user.

The countermeasure to an incorrect implementation of cryptography is to thoroughly verify the cryptographic algorithm (c8) after system implementation.

*7) Side-Channel Attacks*

Side-Channel Attacks: In cryptographic devices such as smart cards, data useful to an attacker other than input data and output data may ' leak out' during cryptographic procedures. The computation time of cryptographic procedures is one kind of such data, as is power consumption. Because the smart card uses an external power source, power consumption can be monitored. Smart cards might be used for authentication purposes only at the client side. In this case, the card and the external power source are both assumed to be secure since they are used by the client and not by the EC system itself.

*B. Security- Oriented Authorization Design Model*

Authorization is the process of giving someone the permission to do or have something.

In multi-user computer systems such as EC systems, a system administrator defines

which users are allowed access to the system and what privileges of use (such as access

to which components, hours of access, and so forth).The specific security attacks related to authorization in e-commerce systems are as follows[6][9][10].

- Session hijacking attacks
- Authorization bypassing attacks
- Privilege brute-force attacks
- Replay attacks
- ID Spoofing attacks

For each authorization attack, derive its enablers and countermeasures This section provides a succinct abstract description of authorization-related security attacks. Attack enablers are then identified and effective countermeasures are prescribed The attacks are presented and discussed below in order of dependence; since some of them are related (e.g. Session hijacking attacks depend on ID spoofing attacks).

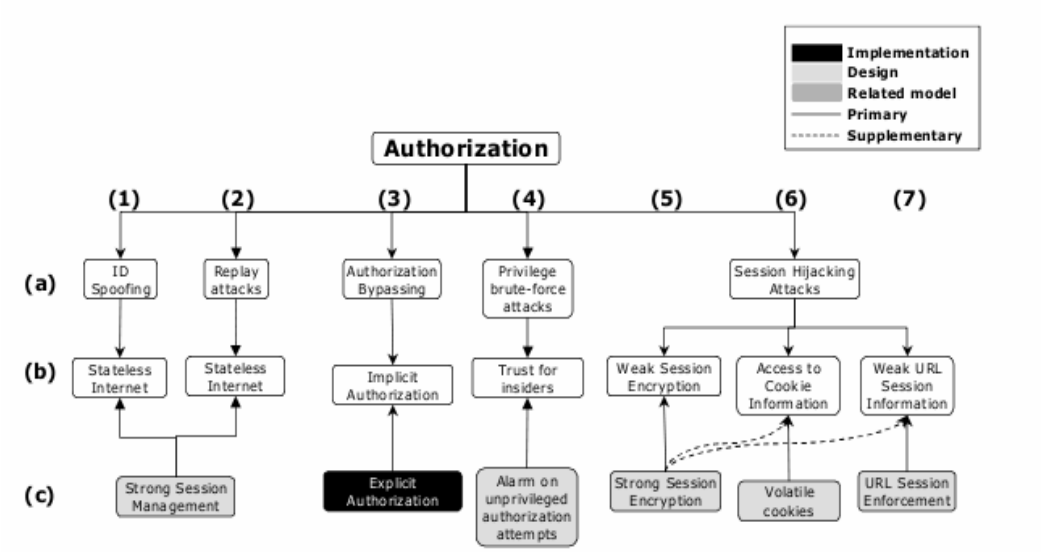


Figure 3: Organizational chart for authorization security attacks(a), attack enablers(b), and countermeasures(c)[6].

*1) ID Spoofing Attacks*

- ID spoofing attacks occur when a malicious user or process claims to be a different user or process (a1).
- The attack enabler for this type of attacks on EC systems is the stateless nature of the Internet part of EC systems (b1). Stateless means that there is no record of previous system interactions and that each interaction request has to be handled based entirely on information provided with it.
- The proper countermeasure is to provide a stateful EC system. Stateful means the system keeps track of the state of interaction, usually by setting values in a storage field designated for that purpose. In other words, stateful means the ability to identify the user across multiple EC system requests. In this case, state information can be kept through the usage of session management techniques (c1).

*2) Authorization Bypassing Attacks*

- Authorization bypassing attacks (a3) might succeed with a weak authorization implementation in place. This attack might succeed especially with systems implementing distributed authorization architectures.
- The attack enabler in this case is that security designers might implicitly rely on the fact that EC system users will not be given access to a certain system component unless properly authorized. This is also known as implicit authorization (b3). As a result, actual authorization might not take place in the system implementation when trying to access the system component and, thus, the authorization process can be bypassed.
- The proper countermeasure for this attack is to enforce explicit authorization (c3) throughout the whole EC system. By doing so, every system component will be forced to explicitly authorize users before giving access. Thus, authorization bypassing attacks are disabled.

*3) Privilege Brute-Force Attacks*

- Privilege brute-force attacks (a4) are similar to brute-force attacks and were discussed earlier in the case of authentication. What is different in this case is that the malicious user is already authenticated and provides a certain level of trust to the EC system, also known as an insider. The goal of this attack is to brute-force the EC system for higher access privileges. An example of such an attack is when a malicious user registers as an ordinary user and uses session management information and system commands to access unauthorized components or parts of the EC system. In such a case, malicious users might be able, through complex tools, to perform reverse-engineering analysis of the EC system. This might lead to more knowledge of the system architecture and, thus, results in an increase in the probability of the attack success.
- The attack enabler in this case is when the system trusts registered users (b4) without taking into consideration the possibility of having a malicious user, also known as an insider.
- The proper countermeasure for this attack is to alarm the EC system administrator when unauthorized access attempts seeking higher privileges take place (c4). Repeated attempts from the same system user will help the system administrator capture a disable the attack either by warning the user or by disabling his EC system account.

*4) Replay Attacks*

- Replay attacks (a2) in the case of authorization are similar to a certain extent to the case of authentication.
- As discussed earlier, the attack enabler for this type of attacks on EC systems is the stateless nature of the Internet part of EC systems (b2).
- The proper countermeasure in the case of authorization is to provide a stateful EC system through the usage of session management techniques (c2). This will prevent malicious users from claiming false identities when attempting to seek authorization.

*5)Session Hijacking Attacks*

- This type of attack involves an attacker using captured, brute forced, or reverse-engineered authorization information (such as session information) to seize control of a legitimate user's session while that user is logged into the EC system. This usually results in the legitimate user losing
- three options for saving session information on the client side: using cookies, URLs, or hidden HTML forms. Whether saved in the URL or in a hidden HTML form, session information is clearly seen by anyone. Thus, for the purpose of our discussion, using URLs and hidden HTML forms are similar.
- The session hijacking attack enabler has three properties: weak session encryption (b5), access to cookie information (b6), and weak URL session information (b7).
- The first attack enabler property is a weak encryption algorithm that allows malicious users to capture session information, decrypt it, and perform session hijacking attacks.
- The countermeasure for this attack enabler property is to have a strong session encryption algorithm (c5). This will prevent malicious users from retrieving useful session information for the purpose of session hijacking attacks in a timely manner.
- The second attack enabler property, access to cookie information, occurs when a session lifetime is longer than the web browser session. Web browsers, in this case, are forced to save cookies on local user hard drives. This allows malicious users, through complex tools, to retrieve saved session information and perform session hijacking attacks.
- The countermeasure for this attack enabler property is to use volatile cookies (b6). Volatile cookies are not saved on local user hard drives. They are saved in the system memory, and once the web browser session ends, i.e. the web browser is closed, the cookie is erased from memory and can not be retrieved.
- The third attack enabler property is using weak URL session information. This usually occurs when cookies are not available and is, practically, similar to

access or functionality to the current EC system session, while the attacker is able to perform all normal application functions with the same privileges of the legitimate user. This type of attacks usually relies on a combination of other simpler session management attacks (such as brute-force attacks and replay attacks). The act of taking control of the session after successfully obtaining or generating an authentication token is called session hijacking. The user may or may not still have all or partial control of his E system session, and may be forced out in the process. An attacker might be able to take control of an active session simply by pasting a URL into his web browser or by loading stolen cookie data and accessing a particular web site or URL (similar to a replay attack). Session management information is usually encrypted and sent to the client side where web browsers save a copy for later usage. There are

saving session information in hidden HTML forms. In this case, session information cannot be volatile since it can be seen by the human eye either in the URL or in the HTML file. Malicious users can, through special tools, retrieve a copy of the session information and start a session hijacking attack.

- The proper countermeasure is to have URL enforcement (c7) This includes but is not limited to re -authenticating the user before critical actions are performed (such as finalizing purchase orders, requesting money transfers, etc.), and mapping session information to web browser instances.

## III. DISCUSSION

The complexity on understanding trust has encourages scholars and researchers to conduct well known and internationally recognized research on numerous trust related issues in the field of e-commerce. Due to the need in explaining and clarifying the concept of trust in e-commerce, various trust related models arises to provide a better and more focused understanding on how trust affects the functional side of e-commerce.

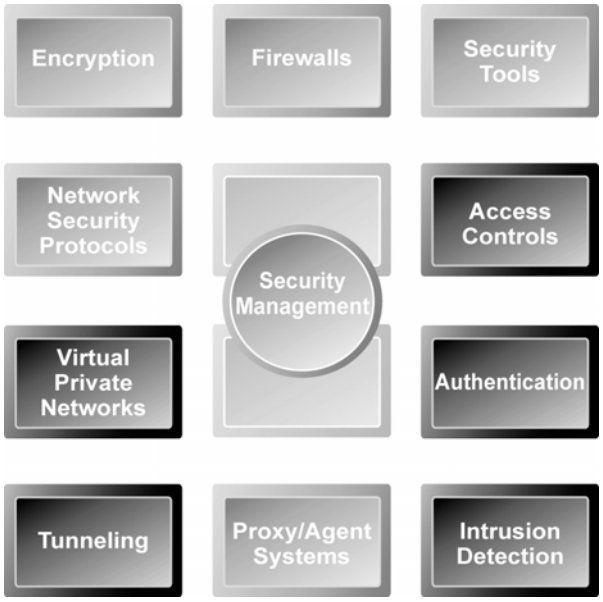


Figure 4: Security areas of E-commerce application

There are following technological and non technological limitations of E-commerce applications.

TABLE I. LIMITATIONS OF E-COMMERCE APPLICATION

| Technological Limitation | Non-technological Limitations |
|---|---|
| 1. There is lack of universal accepted standards for quality, security and reliability. | 1. Security and privacy concerns defer customers from buying. |
| 2. The telecommunications bandwidth is insufficient. | 2. Trust in E-Commerce an in unknown sellers hinders buying. |
| 3. Software developing tools are still evolving | **3.** National and international government Regulations sometimes get in the way. |
| 4. There are difficulties in integrating the internet and E-commerce software with some existing(legacy) applications and databases. | 4. It is difficult to measure the benefits of E-commerce. such as effectiveness of online advertising. There is a lack of mature methodology. |
| 5. Special web servers in addition to the network servers are needed(added cost). | 5. Some Customers like to feel and touch products. Customers are resistant to the change from a real to an online store. |
| 6. Internet accessibility is still expensive and/or inconvenient. | 6. There is sufficient number(critical mass) of sellers and buyers needed for profitable E-commerce |

| | |
|---|---|
| | operations. |

## IV. CONCLUSION

Trust management in e-commerce will remain challenging for some time because the issues and solutions surrounding it are so complex. Security is a requisite and vital concern that should be addressed in e-commerce systems. Commerce and security are inseparable. The reason for wanting to buy, sell, trade, and rent goods is that they are valuable, and valuable items, tangible or intangible, always need protection. Hence the attacks related to authentication and authorization have to be managed by using different countermeasures for better performance of e-commerce application in advance.

## REFERENCES

[1] Dieter Gollmann, E-commerce security, COMPUTING SI CONI'KOL EN(;INB~RIN(;JOUI(NAI. JUNE 2000

[2] Timothy E. Lindquist, Security Considerations for Distributed Web-Based e-commerce Applications in Java, IEEE/2002

[3] PATRICIA BEATTY, IAN REAY, SCOTT DICK, and JAMES MILLER, Consumer Trust in E-Commerce Web Sites: A Meta-Study, ACM Computing Surveys, Vol. 43, No. 3, Article 14, Publication date: April 2011.

[4] Stuart Feldman , The Changing Face of E-Commerce: Extending the Boundaries of the Possible, IEEE/ MAY • JUNE 2000

[5] Vijay Ahuja, Building Trust in Electronic Commerce, IEEE/2000

[6] Thesis by Victor Sawma, *E-commerce Security*, Master of Computer Science, University of Ottawa, Canada 2002

[7] Raj Veeramani and Nancy Talbert, Looking Back at Struggles, Looking Ahead to Opportunities, IEEE/ January ⁞ February 2001

[8] Yacine Atif, Building Trust in E-Commerce,IEEE/2002

[9] Adam Jolly, "*The Secure Online Business*" (Great Britain and the United States- Kogan Page Limited 2003)

[10] Donal O.Mahony, Michael Peirce Hitesh Tewari, "*Electronic Payment Systems for E-Commerce*" (Artech House computer security series-Boston 2001)