DESIGN AND IMPLEMENTATION OF AN INFORMATION SECURITY MODEL FOR E-BUSINESS

Ghasem S. Alijani, James E. Christy, Hugh Craft, Peter Mok, J. Steven Welsh

Southern University at New Orleans
Graduate Studies Program in Computer Information Systems
6400 Press Drive
New Orleans, LA 70126, USA

ABSTRACT

One of the foremost small businesses security concerns is the protection of critical information, both within their internal financial infrastructures and from external elements. Studies show that most cyber-attacks occur inside organizations, instigated by personnel with valid access to the system. This paper describes the design, implementation, and testing of a security system that enhances the capability of small businesses to protect information within the boundary of their networks. Within a specified network, database and transactions are encrypted, decrypted, and processed by the Internal Control and Employee Agents. The database was clustered and access method was provided to employees using private keys. The test results indicate that this additional security layer provides a simple solution to the data sharing and transition within an organization's network. The results of this study will be of significance to owners, managers, and the security personnel responsible for small business networks.

Keywords: Electronic commerce, information security, internal control agent

1. INTRODUCTION

All aspects of E-business and E-commerce have been aided by rapid advances in communication technology and networking (Taylor, 2001 & Baker, 2003). Advertising has been expanded beyond the traditional broadcast media and search engines now facilitate locating businesses faster than ever. Along with these positive advances come major concerns such as security and integrity of information. Hackers can utilize both active and passive methods to overload networks and alter data (Mirza, 2002).

Businesses often implement a variety of "standard" security precautions that give executives and customers a false sense of security (Lenstra, 1996). In a recent survey completed by the Computer Security Institute (CSI), 252 network security professionals reported 202 million dollars in losses due to security breaches and a Symantec survey of 400 companies in 2001 indicated that the

average company experienced 30 electronic attacks per week.

It has been demonstrated that network security applications such as firewalls, password protection, and the inspection of web activities can effectively protect network input (Leonhard, 2000). However, once the data passes through the firewall, it must be protected using different techniques. Also, attacks from within the organization are often difficult to defend since employees may have complete access to customer information (Tribunella, 2002).

If the data is to be protected is within the organization, additional security is needed (Hamann, Henn, Schack, & Seliger, 2001). This additional layer of security must be mission-oriented and customizable in order to both meet business requirements and allow data sharing within the network boundary. The objective of this research work was to develop and implement a model that enhances the internal information security of small businesses. The model utilizes the RSA (named after its developers, Rivest, Shamir, and Adleman) algorithm for encrypting, transmitting, and decrypting sensitive data being shared among the employees within a network.

2. BACKGROUND

Using various hacking techniques, intruders can gain access to an unsecure system from any location. The probability of identifying attackers is very minimal -- 1 in 20,000 according to FBI statistics (Gabrys, 2002). Following are a few protection alternatives.

Password Protection. Passwords are utilized to control access to a system or network. However, for a password to be effective it must be complicated and be changed often so that it cannot be used extensively if discovered. Unfortunately, users often fail to follow these simple guidelines. According to Armstrong (2003), memorization of many passwords is difficult, if not impossible. Another alternative that may strengthen security is digital signatures. These signatures can be sent via the public key infrastructure and combined with smart cards and a personal identification number (PIN) (Kuechler and Grupe, 2003).

Protection of Network Boundary and Connections. A firewall can protect a network from external attacks by examining all packets of a message attempting to pass through the network and rejecting the packets that do not meet the security restrictions (Moon et al., 2003). However, it does not protect the data as it is transmitted from one network to another.

Data transmitted from one network to another via the Internet is susceptible to access at many points between the source and destination. The secure socket layer (SSL) is one means of providing secure communications between points connected via the Internet (Kant & Mohapatra, 2000). Busta (2002) describes the basics of SSL, which uses a combination of public and private keys for encryption and hashing to secure the data. SSL ensures the integrity of data being sent and limits data access to the intended receiver (Apostolopoulos, Peris, Hradhan, & Saha, 2000).

Encryption and Decryption. A successful encryption process renders captured data useless to hackers. There are several symmetric encryption algorithms, including the Data Encryption Standard (DES), the Triple Data Encryption Algorithm (TDEA), the

International Data Encryption Algorithm (IDEA), the Blowfish algorithm, the RC5 algorithm, and the CAST-128 (Stallings, 2000).

As with symmetric encryption, public key encryption can utilize several different algorithms. The most popular public method is RSA. It creates a public key and passes it to the data sender. RSA also creates a private key which the receiver uses to decipher the message. Since the private and public keys are different, the receiver need not divulge the private key to anyone (Poddar, Singh, Vinoo, & Saraswat, 2003). Other popular public key encryption algorithms include Diffie-Hellman, Digital Signature Standard (DSS), and Elliptical-curve cryptography (ECC).

3. METHODOLOGY

Commercial off-the-shelf security software may not always provide the level of security anticipated. White and Alijani (2003) examined products from 186 vendors and found that only 11 were compliant with 75 percent of the study requirements and only one provided all the required security features. An alternative is a mission-oriented security model that can be modified according to custom specifications. The intention of this research work was to illustrate that such an additional layer of customized security can protect data within an organization's network. The advantage of such a customized solution is that the Employee Agent (EA) can control the key generation parameters and determine the frequency for regenerating public and private keys. Following figure shows physical structure of the system.

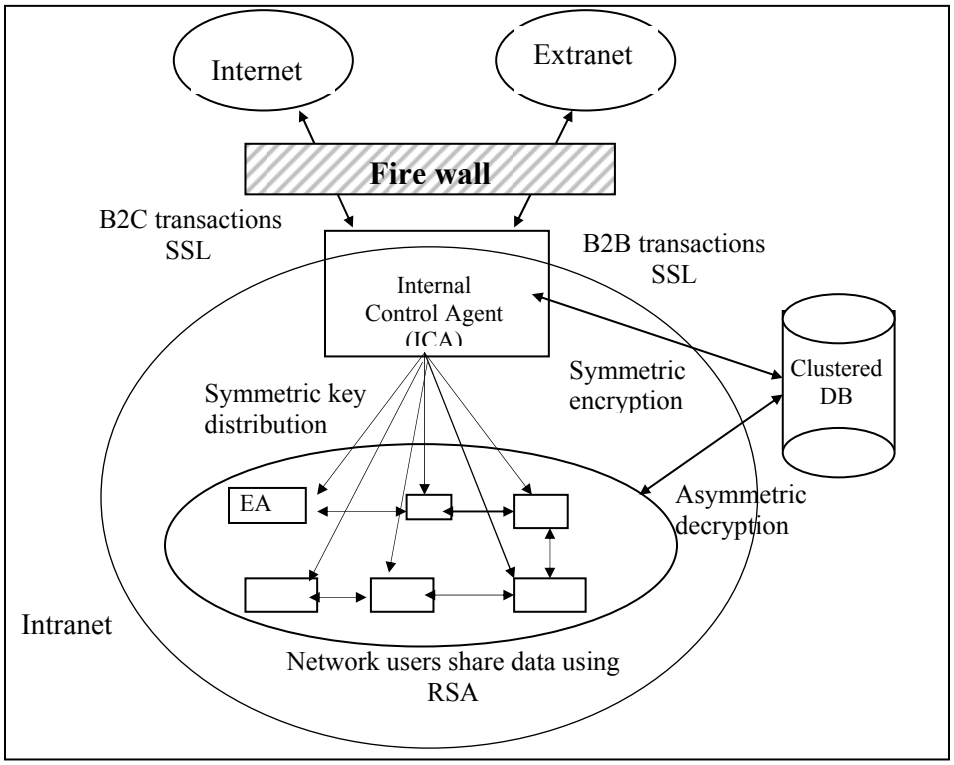

Figure 1 Physical Structure of Data Security System for a Small Business

3.1 Functional Specification

Within the specified network, the ICA and the EAs are the primary active components and their functions are described below.

ICA: The ICA provides restricted access to each cluster of database to the authorized EA. This is accomplished by encrypting the data prior to writing to the database. Each cluster of data is encrypted using a different key, so that only the authorized workstation can gain access to the information. The ICA could exist in one of the following states:

Receiving State

1. In this state, the ICA accepts customer and business data transmitted through the firewall from the Internet or extranets.
2. The ICA also determines which secret key to utilize for symmetric encryption of data.

Processing State

1. When the ICA is in this state, secret keys are generated using symmetric key encryption. The keys are used to store customer and business account data into a database.
2. The ICA also manages distribution of the keys, based on employee account responsibility. Accounts may be assigned to the EAs by various means, such as geographical location or account balance.

Sending State

1. In this state, the ICA distributes secret keys to the appropriate EAs.
2. The ICA encrypts data that is received from external networks using an appropriate secret key.
3. The ICA writes data to the clustered database.

EA: Employees manage the business accounts from their own workstations. The sharing of data between EAs is protected by the RSA security layer. As with the ICA, each EA could exist in one of three states:

Receiving State

1. In this state the EA reads encrypted files.
2. The EA decrypts the ciphertext using the private key.
3. The EA also displays decrypted data onto the screen.

Processing State

1. When the EA is in this state, it performs day-to-day business functions.
2. The EA also generates public and private keys using the RSA algorithm.

Sending State

1. In this state the EA validates data to be delivered.
2. The EA encrypts data using the public key of the destination workstation.
3. The EA also writes the encrypted ciphertext files.

To share data with other employees, the sender encrypts data using the receiver's public key and the receiver decrypts the data using the associated private key. This additional layer of security protects data from internal users that do not have valid access.

3.2 Design Analysis

Design analysis includes the inputs, processes, and outputs of the active elements which are the ICA and EAs.
*Inputs***.** Table 1 identifies the data to be inputted when the EA is in the sending state.

Table 1
Sending Process Input Data

Data Field	Value Constraints
UserId	8 characters
Password	8 characters
CreditCardName	Selected from a dropdown box
CreditCardNumber	Each of 4 segments contain 4 numeric characters
ExpirationDate	Selected from a dropdown box
PublicKeyE	Created using RSA algorithm
PublicKeyN	Created using RSA algorithm

With the exception of the public key, the user inputs all the data, which must be validated. Once validation is achieved, the data is encrypted using the public key. The display process of the workstation operates in the receiving state. This process decrypts cipher data created in the data input module for display to the user. Table 2 identifies the inputs for this Process.

Table 2

Identification of Display Process Inputs

Data Field	Value Constraints
PrivateKeyD	Input from RSA key creation process for decryption.
PrivateKeyN	Input from RSA key creation process for decryption.
Cipher text	Input from data input process

Processing. This system includes three major steps:

1. The receiver generates public and private keys using the RSA algorithm.
2. The sender encrypts sensitive data entered through the interface, using the public key created by the first steps.
3. The receiver decrypts and displays data that was encrypted in the previous step. (Decryption uses the private key created in first process.)

Outputs. There are three outputs when the workstation is in the sending state.

1. After the key generation is complete, the workstation writes the public key to a shared location.
2. The workstation writes the private key to a protected location.
3. When the data validation is complete, the workstation writes the ciphertext file.

The Business Process Model (BPM) provides a high level view of the RSA model. This RSA model requires three major processes. The first is to utilize the

RSA algorithm to create the public and private keys and distribute the public key. The second process involves entry of the sensitive data, including editing, encrypting, and transmitting. The third process decrypts the cipher text and displays the information at the workstation.

The implementation of the model includes three graphical user interface (GUI) screens. The sequence of the entire process is presented by screens S1, S2 and S3 discussed below. Screen S1 provides the user with the capability to generate a new public and private key utilizing the RSA algorithm. The private key can then be distributed for use by the entity responsible for entering and encrypting the data

Screen S2 provides the user with data input capability. When the user selects the submit button, the data is edited for validation. The employee is then prompted for correction. After validation, the module encrypts the data and the cipher text is saved to a file.

Screen S3 provides the receiving EA with the ability to decrypt and view the data, using the private key created at the destination workstation. Notice that if the receiving EA generates new keys prior to decryption, then the sending EA needs to re-encrypt the data using the new public key.

3.3 Key Generator Process

The key generation process uses the RSA algorithm to create the public and private keys for encryption and decryption, respectively. A random number generator is used so different input values are derived every time the program is executed. The outputs of this component are the public and private keys. The steps required to generate each key are as follows:

1. Generate prime numbers p and q randomly.
2. Find variable n = p*q.
3. Find (p-1)(q-1) and store in variable Φ(n), known as the Euler Phi(Φ) function.
4. Select variable e, such that the greatest common devisor (GCD) between e and Φ(n) is equal to 1, and e is between 1 and Φ(n).
5. Calculate variable d, such that $d = e^{-1} \bmod \Phi(n)$.
6. Save the public key, KU={e,n}, in a location accessible by sending EA.
7. Save the private key, KR = {d,n}, for future decryption.

3.4 Encryption Process

This process accepts and edits the user input or it reads the data from customers' databases. It then validates and encrypts the input message. The inputs include the public encryption key and the data items identified as user input in the system specifications. The encryption process involves the following activities:

1. Convert each character into its ASCII numeric equivalent.
2. Encrypt plaintext data M, to create cipher text C. Encryption occurs one character at a time using the formula $C = M^e \pmod n$. The result of the exponentiation is a number that is too large for storage in a 32-bit register. To correct this, if e is greater than n, then e mod n will be

used as a new e.

3. Save and transmit the cipher text.

When the entire message is encrypted, it is written to a sequential file as ciphertext.

3.5 Decryption Process

This process decrypts the data and displays it on the screen. The input is the ciphertext and the private key produced by the key generator and encryption processes, respectively. The decryption and display process involves the following activities:

1. Read the cipher text data file.
2. Load each character of cipher text into an array.
3. Decipher individual characters C to discover original message M, using the RSA decryption formula, $M = C^d \pmod{n}$. The problem of very large numbers applies and is solved the same as for encryption.
4. Convert the numeric ASCII representation to its original character format.
5. Store the encrypted characters into the original data fields.
6. Display information on the interface screen.

4. IMPLEMENTATION

In addition to the Graphic User Interface (GUI), each EA is facilitated by three distinct processes: key generation, encryption, and decryption. From the main menu, the receiving site (an EA) can generate the keys or select the display menu to view the received data. When the employee selects the "Regenerate Keys" button, the system generates the public and private keys and saves them in two separate files. For testing purposes, the system writes the files onto an external storage device. The private key is saved at a location with restricted access for all other users. The employee can terminate the program if he/she selects the "Exit" option. When the employee selects the "Display Menu" button, he/she can view the data. Figure 2 illustrates the main menu after the employee regenerates the public and private keys.

Before the receiving workstation can review a message, the sender must have used the correct key to create an encrypted message. For the purpose of validation, the password that the user provides is not hidden by asterisks. Once the "submit" button is selected and the data is validated, the public key file is read and the data is encrypted.

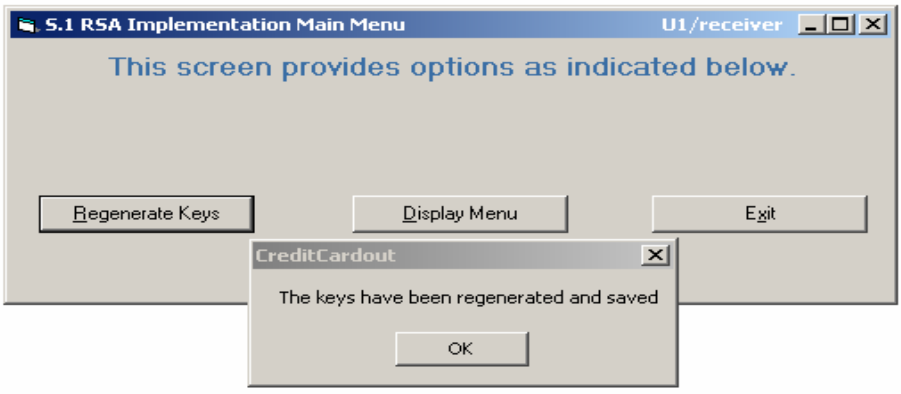

Figure 2: Main Menu for User 1 – Receiving State

The encrypted file is written to external storage media for testing. The sending workstation encrypts and saves the data as illustrated in Figure 3.

the sending workstation. The receiver then reads the private key produced during the key generation process and the data is decrypted and displayed at the workstation. Once the sender saves an

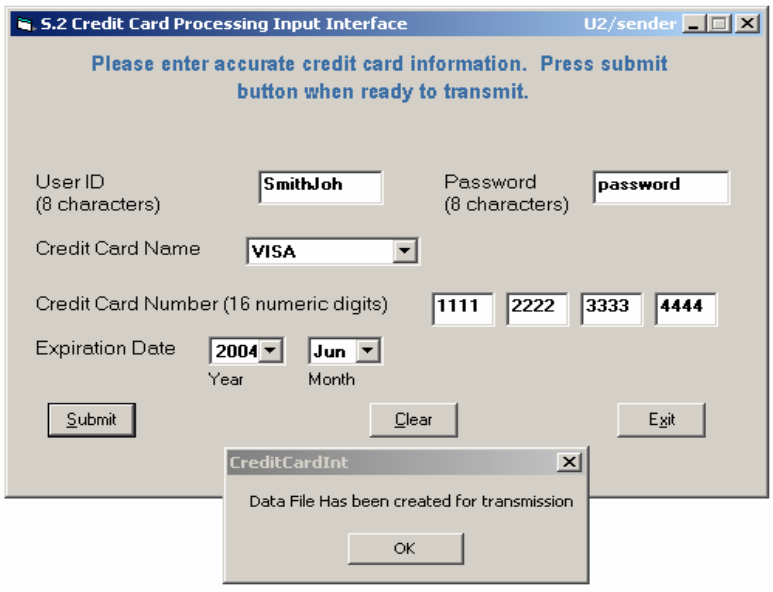

Figure 3: Operation of the Data Entry Module for User 2 –Sending State

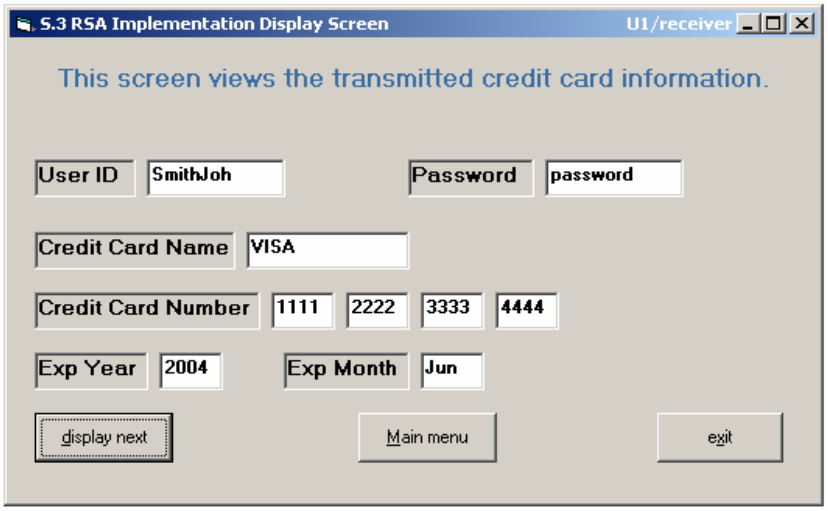

Figure 4: Data Display Process for User 1 – Receiving State

When the receiving workstation selects the "display next" button, the system reads the file that was written by

encrypted file, the receiver can view the formatted information using the interface illustrated in Figure 4.

5. EVALUATION

5.1 Process Testing

Upon completion of the system's development phase, multiple tests were conducted to ensure that each process and the entire system was operating properly. Multiple keys were generated to ensure that the algorithm worked properly. The tests are repeated with different input and public/private key combinations. The tests also included detection and correction of abnormal program termination, as well as incorrect results. The results generated by the application program were displayed on a run-time basis. Table 3 illustrates the results of a small sample of the numerous successful tests. It displays the keys generated at the receiving workstation, the data input at the sending workstation, and the data outputted at the receiving workstation.

5.2 System Parameter Testing

At the network level, tests are done using 100 pre-determined test scripts, which were run through the encryption and decryption processes to determine any abnormality in the parameter values. Each of the test scripts represents a secure exchange of a customer record between two EAs. Each repetition of a test represents an option where an EA changes the security parameters and generates a new key. Each of the 100 scripts are encrypted and decrypted five times using five different random keys -- a total of 500 different keys. For each test, a total of 43 characters were verified and compared on an all-or-none basis after the encryption, transmission, and decryption process. Due to space restrictions, only 4 customer records are shown in Table 4 and Table 5, which depict a snapshot of the system after the matching process.

Table 3

Results of Process Correctness Test

Test No.	Public Key KU{e,n}	Private Key KR{d,n}	Dada Character String I = Input C = Ciphertext O = Output	
1	{3, 493}	{299, 493}	I	AB1234YZPASSWORDVISA1234567890123456 2004Jan
			C	□;□□□"gØ□Z□□□□□□□\□'Â‡□V(□□□;□□□"gä□Åêl >□;□□□"gä□Å□□g×□,‰□
			O	AB1234YZPASSWORDVISA1234567890123456 2004Jan
2	{7,427}	{104, 427}	I	JOHNSMITPASS1234MAST888877776666333320 05Feb
			C	J/_é__,___¤bøÛ,_,_□_<_'_Z___%___*_MMMMííí íÉÉÉÉ'_'_'_'_<_00M_Ë□
			O	JOHNSMITPASS1234MAST888877776666333320 05Feb
3	{3, 711}	{1083, 1711}	I	BobAppleWater000AMER945387564537237820 08Dec
			C	0__Ža_ÅÅ___Ç_Æ_Ð___ù_@_@_@_a_é___ù_•_2__^ _F_~__42__^_~__42__a@_@_F_'___¢
			O	BobAppleWater000AMER945387564537237820 08Dec

Table 4
Sample Scripts (customer records)

Test Script Number	User ID	Password	Credit Card Name	Credit Card Number	Exp Year	Exp Month
1	AB12#$gH	34!?Dkl:	VISA	1352877692465710	2004	Jan
2	CdPo^"[}	QM,\CfvL	Master card	9661749000658321	2005	Feb
3	<@2HkBZ4	A(PTxJG#	Discover	5432123454321234	2006	Mar
4	W!E$r%y&	Allsmall	American Express	9876678998766789	2007	Apr

Table 5
Sample Test Results

Script No	Public Key KU{e,n}	Private Key KR{d,n}	Chars Matched	Script No	Public Key KU{e,n}	Private Key KR{d,n}	Chars Matched
1	{3, 4717}	{3051, 4717}	43/43	3	{5, 2183}	{1253, 2183}	43/43
	{5, 1501}	{281, 1501}	43/43		{3, 2323}	{1467, 2323}	43/43
	{11, 899}	{611, 899}	43/43		{5, 5141}	{1997, 5141}	43/43
	{5, 2923}	{1685, 2923}	43/43		{7, 22}	{3, 22}	**0/43****
	{5, 949}	{173, 949}	43/43		{5, 1591}	{605, 1591}	43/43
2	{5, 185}	{29, 185}	43/43	4	{3, 2047}	{1291, 2047}	43/43
	{7, 3233}	{1783, 3233}	43/43		{3, 187}	{107, 187}	43/43
	{5, 4817}	{3245, 4817}	43/43		{5, 1343}	{749, 1343}	43/43
	{7, 1517}	{823, 1517}	43/43		{11, 627}	{2291, 2627}	43/43
	{5, 6557}	{5117, 6557}	43/43		{5, 8633}	{5069, 8633}	43/43

** unmatched case

6. DISCUSSION

As Table 5 shows, there is one unmatched case, which indicates that the system did not decrypt the message. In 500 test cases, there were 14 (approximately 3%) unmatched cases where the value of ***e*** was greater than or equal to the value of ***d*** and the value of ***n*** less than 100. For a system with a 32-bit register, it is recommended that the value of prime numbers for **p** and **q** be less than 100. However, they should not be too small since the cause of the exceptionally small key values is directly related to the small values for **p** and **q**. To overcome this anomaly, the random number generators can be changed to ensure a minimum value for **p** and **q**.

7. CONCLUSION

The objective of this research work was to design and implement a model which enhances internal security of small businesses. The model utilized the RSA algorithm, along with a clustered database, to encrypt, transmit, and decrypt sensitive data being shared among employees within an organization's network. The test results indicated that

this additional security layer provides a simple solution to the security of internal information processing of small businesses and can be used in tandem with the existing system security layers.

8. ACKNOWLEDGEMENT

This research was partially supported by the National Science Foundation under the PESMaCT Grant HRD-0102620.

9. REFERENCES

Apostolopoulos, G., Peris, V., Pradhan, P., & Saha, D. (2000). Securing electronic commerce: Reducing the SSL overhead. *Network, IEEE* 14(4), 8-16.

Armstrong, I. (2003). Passwords exposed: Users are the weakest link. *SC Magazine,* 14(6), 26-29.

Baker, G. (2003). Applications galore! *New Zealand Management,* 50(10), 50-56.

Gabrys, E. (2002). The international dimensions of cyber-crime, part 1. *Information Systems Security,* 11(4), 21-32.

Hamann, E., Henn, H., Schack, T., & Seliger, F. (2001). Securing e-business applications using smart cards. *IBM Systems Journal,* 40(3), 635-647.

Kant, K., Iyer, R., & Mohapatra, P. (2000). Architectural impact of secure socket layer on Internet servers. *Computer Design, 2000, International Conference, 7-14.*

Kuechler, W., & Grupe, F. H. (2003). Digital signatures: A business view. *Information Systems Management,* 20(1), 19-28.

Lenstra, A. K. (1996). Securing the net – the fruits of incompetence. *First Monday.* Retrieved January 25, 2004, from http://www.firstmonday.org/issues/issue4/lenstra/index.html

Mirza, D. R., Dubrawsky, I., Flynn, H., Grand, J., Graham, R., Johnson, N. L., et al. (2002). *Hack Proofing Your Network, Second Edition.* Rockland, MA: Syngress Publishing.

Moon, J., Cho, H., Choi, C, Jung, G, Younkwang, J., & Choi, K. (2003). Accelerating firewall. *SAM '03 International Conference, 433-436.*

Poddar, V., Singh, V. K., Vinoo, A. E., & Saraswat, P. (2003). *Cryptography Protocols and Algorithms.* Nashua, NH: Skillsoft Press.

Stallings, W. (2000). *Network Security Essentials.* Upper Saddle River, NJ: Prentice Hall.

Taylor, T. (2001). Thinking about a new economy. *Public Interest,* 143, 3-19.

Tribunella, T. (2002). Twenty questions on E-commerce security. *CPA Journal*, 72(1), 60-63.

White, D., & Alijani, G. (2003). Identifying requirements for network security software. *SAM '03 International Conference,* 539-543.