SFAMSS:A SECURE FRAMEWORK FOR ATM MACHINES VIA SECRET SHARING

Zeinab Ghafari[1], Taha Arian[1] and Morteza Analoui[2]

[1]School of Computer Engineering Iran University of Science and Technology, Tehran, Iran
[2]Associate Professor of Computer Engineering Iran University of Science and Technology, Tehran, Iran

ABSTRACT

As ATM applications deploy for a banking system, the need to secure communications will become critical. However, multicast protocols do not fit the point-to-point model of most network security protocols which were designed with unicast communications in mind. In recent years, we have seen the emergence and the growing of ATMs (Automatic Teller Machines) in banking systems. Many banks are extending their activity and increasing transactions by using ATMs. ATM will allow them to reach more customers in a cost effective way and to make their transactions fast and efficient. However, communicating in the network must satisfy integrity, privacy, confidentiality, authentication and non-repudiation. Many frameworks have been implemented to provide security in communication and transactions. In this paper, we analyze ATM communication protocol and propose a novel framework for ATM systems that allows entities communicate in a secure way without using a lot of storage. We describe the architecture and operation of SFAMSS in detail. Our framework is implemented with Java and the software architecture, and its components are studied in detailed.

KEYWORDS

ATM Security, Framework, Secret Sharing, Authentication, Protocol Design, Software Architecture.

1. INTRODUCTION

Security protocols are applied on untrusted network to enhance their safety. ATM networks are one of the typical networks that need a high level of security to prevent the attacker doing malicious activity. ATM communication consists of several phases, such as authentication and authorization. So, establishing a comprehensive security in ATM infrastructure needs a lot of concerns. Lack of security even in one of the phases can lead to massive security breach. In order to solve this problem, we propose a new framework that includes several entities same as ATM, customer, and bank. To achieve security in the ATM networks each entity should consider the security as an important factor.

There are some studies on designing a new protocol and their attacks on ATM. In our previous paper [1] we presented two security protocols for ATM communication. In this paper, our motivation is to introduce a new framework that includes registration, authentication, and authorization. Firstly, the user and ATM register for bank's services. The bank generates authentication information and distributes it among them. Bank also assigns some privileges to users, such as the amount of money he could transfer by means of ATM. Secondly the bank authenticates user by ATM, and finally the bank authorizes user's request for a service, based on the user's privileges. It is possible that a user be authenticated by the bank; however, he is not authorized for a particular service. We implement our framework by using Java programming

DOI:10.5121/ijcsit.2015.7206

language on the Eclipse platform. Our framework will investigate as a secure framework for ATM machines via secret sharing (SFAMSS) in the rest of paper.

Secret sharing schemes (SSS) are perfect for storing information that is highly confidential and critical. For instance encryption keys, missile launch codes, and bank account information are critical information that should maintain in a secure way. This information must be protected highly confidential, as their exposure could be harmful. Traditional methods for encryption are not suited for achieving high levels of confidentiality. Secret sharing aims at efficiently sharing a secret among a number of entities, and the secret can be recovered by entities shares. Hence, there are several applications of secret sharing in computer science such as [2],[3] in cloud and [4],[5] in data outsourcing.

The rest of the paper is organized as follows: Section 2 analyzes related work and Section 3 describes the framework in particular, highlighting its main architecture and protocol details. Then, Section 4 illustrates the implementation, and a case study in detail. Section 5 reports the protocol analysis. Finally, Section 6 concludes the paper.

2. RELATED WORKS

There is a considerable amount of literature on designing a new secure framework in network communication. Few studies have been published on banking systems. In [6] the authors proposed a framework based on a smart card that allows entities to realize secure transactions. The proposed solution uses smart cards to store keys and perform cryptographic algorithms in e-business transactions. Robinson P. et.al. [7] reported a new framework for a secure protocol. The paper describes the implementation of a deterministic and fair non-repudiation protocol. Malin and Sweeney [8] presented the STRANON protocol that enables locations to cooperate such that health records are not revealed. They developed a general high-level framework for STRANON. Authors in [9] presented a new framework intended to extract FSA(Finite State Automaton) specifications of network protocol implementations and test it for implementation flaws. They constructed the framework using Java. Mittra [10] developed a novel framework for scalable secure multicasting this protocol can be used to achieve a variety of security objectives in communications. In [11] the authors investigated a new framework for network management protocol. They proposed an architecture composed of mobile agents. Camenischet.al[12] proposed a framework for cloud security storage. They applied secret sharing scheme to prevent unauthorized access to cloud data. They provide a secure analysis for their protocol by using zero-knowledge proof.

3. SFAMSS FRAMEWORK

In this paper, we implement the authentication and secure communication framework for a distributed ATM systems, consisting of a bank server, a number of ATMs and users. Since the computer system at the bank is connected through an insecure communication channels to the ATM, applying secure mechanisms in the communication is essential. The insecure communication channels are subjected to attacks by active and passive malicious attacker. Messages may not be removed by an attacker. However, confidentiality of messages might be breached, and new message might be generated by the attacker. Thus, the authorship and content of messages that transit in the insecure communication link should be considered suspect. Customers use ATMs to make queries such as withdrawals and balance inquiries involving their accounts. Attackers must be prevented from interfering with these actions.

Communications between user, ATM and bank consists of the following steps:

1. The user inserts his smart card into the ATM.
2. The ATM informed the customer for a password.

3. The user selects a request to be completed by ATM, and that request is performed by the bank and the result is shown by the ATM to the user.

3.1. SFAMSS Architecture

In this section, we considered that three parties in ATM system do not trust each other among interaction network. So, to protect interacts among parties, communication should be confidential and reputable. Confidentiality is the property that a message cannot be accessed by unauthorized entities. Non-repudiation is the property that an entity cannot repudiate the message that he send before. In this section, we present a new protocol that guarantees non-repudiation and confidentiality. Figure 1 illustrates SFAMSS architecture with SSS component that changes the authentication process of ATM systems. Using SSS helps entities to authenticate each other without trusted third party existence.

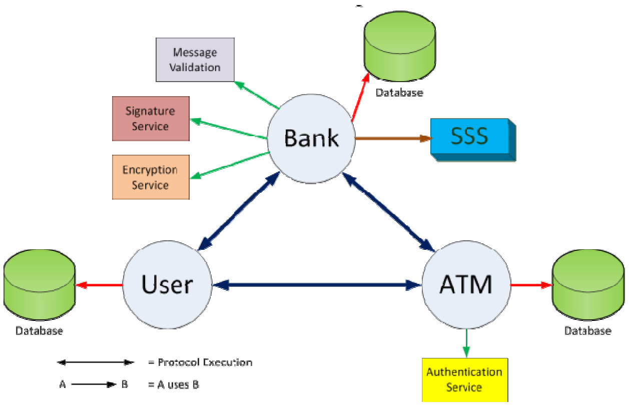

Figure 1-SFAMSS architecture

3.2. Protocol details

We have a certificate authority that issues certificates to all entities, so, before the registration phase all of the entities have their certificate and private key. In all of the protocol descriptions, E(key_{AB})(msg) means message (msg) is encrypted by key (key_{AB}). E_{pu}, E_{pr} are referred to entity E public key and private key. K_S is a session key which is established between user and bank. The ATM communicates with the bank by means of a protocol that meets the following requirements:

1. ATM authenticates the user.
2. ATM sends authentication information to the bank.
3. It preserves the integrity and confidentiality of communications between the bank and ATM.

Messages that are sent by the ATM to the bank provide evidence that every ATM-initiated action was, in fact, initiated by the user. The bank server responses to user by ATM. It also records information in an audit log for later use in justifying its past actions to the user. The audit log that is stored in bank servers is vulnerable to network attacks. So data written to the log should be encrypted to prevent a confidentiality breach. In addition to log, bank should keep accurate information about users and ATMs secretly. Since entities sign their messages, they cannot repudiate it.

$bank_{pb}$: Banks public key
$bank_{pr}$: Banks private key
K(u_1; bank): Session key between bank and u_1
u_{1pb} : Users public key

u_{1pr} : Users private key The private key of the user is stored securely in the credit card
ATM_{pb}: ATMs public key
ATM_{pr}: ATMs private key. The private key of the ATM is stored securely at the ATM.

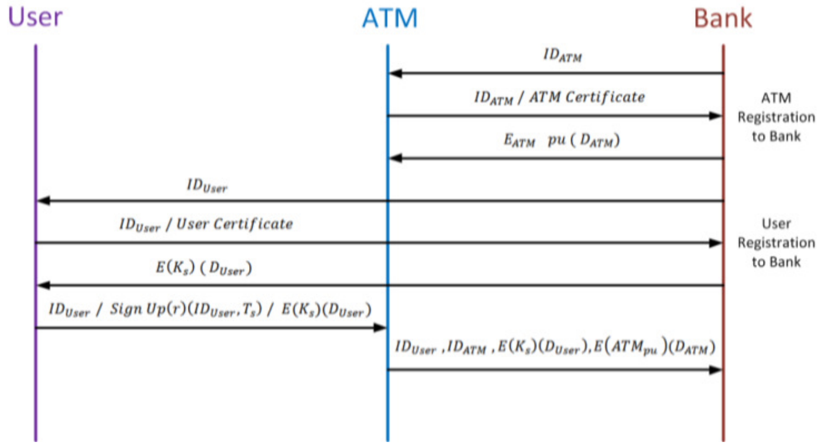
User
ATM
Bank
ID_{ATM}
ID_{ATM} / $ATM\ Certificate$
$E_{ATM}\ \ pu\ (D_{ATM})$
ATM Registration to Bank
ID_{User}
ID_{User} / $User\ Certificate$
$E(K_s)\ (D_{User})$
User Registration to Bank
ID_{User} / $Sign\ Up(r)(ID_{User}, T_s)$ / $E(K_s)(D_{User})$
$ID_{User}, ID_{ATM}, E(K_s)(D_{User}), E(ATM_{pu})(D_{ATM})$

Figure 2-Message Exchange in the selected scenario

Our scenario is illustrated in
Figure and we will investigate it step by step.

3.2.1. Registration of the ATM with the bank

$$BANK\ \rightarrow ATM:\ ID_{\ ATM}$$

$$\mathrm{ATM}\ \rightarrow \mathrm{BANK}: \mathrm{ID_{ATM}}, \mathrm{ATM_{certificate}}$$

$$[\mathrm{BANK\ generate\ function\ F(x)}\&\ calculate\ \mathrm{D_{ATM}} = (\mathrm{ID_{ATM}}, \mathrm{F}(\mathrm{ID_{ATM}}))\quad]$$

$$\mathrm{BANK} \rightarrow \mathrm{ATM} : \mathrm{E}(\mathrm{ATM_{PU}})(\mathrm{D_{ATM}})$$

3.2.2. Registration of the User with the bank

$$\mathrm{BANK}\ \rightarrow \mathrm{USER}:\ \mathrm{ID_{USER}}$$

$$\mathrm{USER}\ \rightarrow \mathrm{BANK}:\ \mathrm{ID_{USER}}, \mathrm{USER_{Certificate}}$$

$$[\quad \mathrm{BANK\ generate\ \ random\ r_{USER}\ \ and\ store\ D_{BANK,USER}} = (0, \mathrm{r_{USER}})\&$$
$$\mathrm{calculate\ f_{new}(x) = f(x) + r_{USER}}$$
$$\&\mathrm{D_{USER}} = (\mathrm{ID_{USER}}, \mathrm{f_{new}}(\mathrm{ID_{USER}}))\quad]$$
$$\mathrm{BANK} \rightarrow \mathrm{USER}:\ \mathrm{E}(\mathrm{K_S})(\mathrm{D_{USER}})$$

3.2.3. Authentication of User to ATM

$$\mathrm{USER}\ \rightarrow \mathrm{ATM}:\ \mathrm{ID_{USER}}, \mathrm{SIGN}(\mathrm{USER\ pr})(\mathrm{ID_{USER}}, \mathrm{T_S}), \mathrm{E}(\mathrm{Ks})(\mathrm{D_{USER}})$$

$$\left[\begin{array}{c}\text{ATM requests user certificate from BANK and}\\ \text{after that ATM verify the signiture}\\ \text{by the user bublic then ATM authenticate USER}\end{array}\right.$$

3.2.4. Authentication of ATM and User to Bank

$$\mathrm{ATM} \rightarrow \mathrm{BANK}\text{: } \mathrm{ID_{USER}}, \mathrm{ID_{ATM}}, \mathrm{E(K_S)(D_{USER})}, \mathrm{E(ATM_{PU})(D_{ATM})}$$

Bank decrypts $\mathrm{D_{USER}}$ with session key betweenBank and User and decrypts $\mathrm{D_{ATM}}$ with $\mathrm{ATM_{PU}}$

and calculate $\mathrm{D'_{ATM}} = (\mathrm{ID_{ATM}}, \mathrm{F(ID_{ATM})} + \mathrm{r_{USER}})$
generate $\mathrm{F'_{NEW}(x)}$with $\mathrm{D_{USER}}, \mathrm{D'_{ATM}}, \mathrm{D_{BANK}}$

$$\mathrm{F'_{NEW}(x)} = ?\ \mathrm{F_{NEW}(x)}$$

4. IMPLEMENTATION

The UML design goal is to represent the several components and technologies that will be used to build a framework. We implemented protocol role with Java language, in the eclipse platform. The UML diagram is illustrated in this hyperlink.

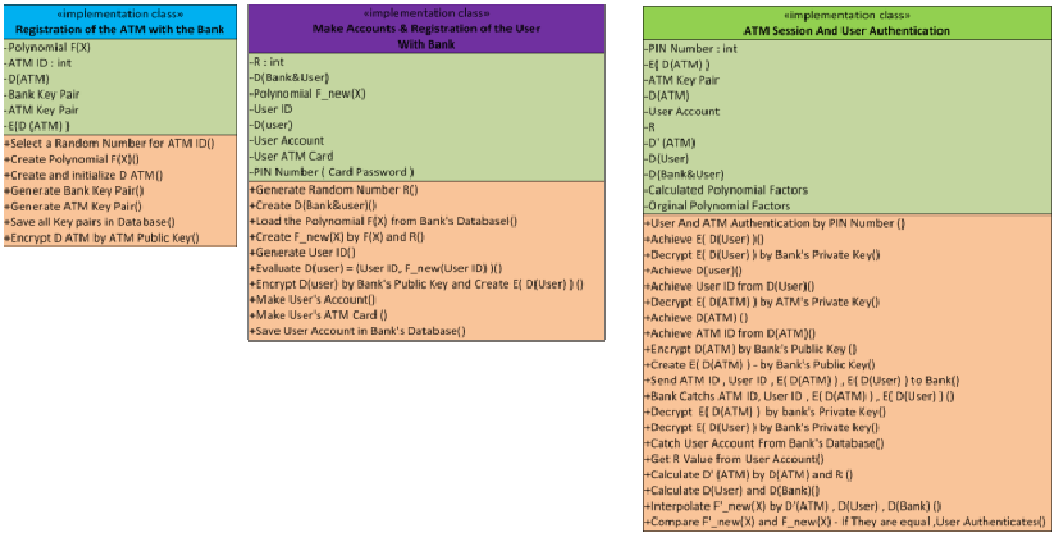
«implementation class»
Registration of the ATM with the Bank
-Polynomial F(X)
-ATM ID : int
-D(ATM)
-Bank Key Pair
-ATM Key Pair
-E(D (ATM))
+Select a Random Number for ATM ID()
+Create Polynomial F(X)()
+Create and initialize D ATM()
+Generate Bank Key Pair()
+Generate ATM Key Pair()
+Save all Key pairs in Database()
+Encrypt D ATM by ATM Public Key()
«implementation class»
Make Accounts & Registration of the User With Bank
-R : int
-D(Bank&User)
-Polynomial F_new(X)
-User ID
-D(user)
-User Account
-User ATM Card
-PIN Number (Card Password)
+Generate Random Number R()
+Create D(Bank&user)()
+Load the Polynomial F(X) from Bank's Database()
+Create F_new(X) by F(X) and R()
+Generate User ID()
+Evaluate D(user) = (User ID, F_new(User ID))()
+Encrypt D(user) by Bank's Public Key and Create E(D(User)) ()
+Make User's Account()
+Make User's ATM Card ()
+Save User Account in Bank's Database()
«implementation class»
ATM Session And User Authentication
-PIN Number : int
-E(D(ATM))
-ATM Key Pair
-D(ATM)
-User Account
-R
-D' (ATM)
-D(User)
-D(Bank&User)
-Calculated Polynomial Factors
-Orginal Polynomial Factors
+User And ATM Authentication by PIN Number ()
+Achieve E(D(User))()
+Decrypt E(D(User)) by Bank's Private Key()
+Achieve D(user)()
+Achieve User ID from D(User)()
+Decrypt E(D(ATM)) by ATM's Private Key()
+Achieve D(ATM) ()
+Achieve ATM ID from D(ATM)()
+Encrypt D(ATM) by Bank's Public Key ()
+Create E(D(ATM)) - by Bank's Public Key()
+Send ATM ID , User ID , E(D(ATM)) , E(D(User)) to Bank()
+Bank Catchs ATM ID, User ID , E(D(ATM)) , E(D(User)) ()
+Decrypt E(D(ATM)) by bank's Private Key()
+Decrypt E(D(User)) by Bank's Private key()
+Catch User Account From Bank's Database()
+Get R Value from User Account()
+Calculate D' (ATM) by D(ATM) and R ()
+Calculate D(User) and D(Bank)()
+Interpolate F'_new(X) by D'(ATM) , D(User) , D(Bank) ()
+Compare F'_new(X) and F_new(X) - If They are equal ,User Authenticates()

Figure 3. Implementation classes

4.1. Case study

In the registration phase of ATM and the bank, first bank generates a key pair for ATM based on the RSA algorithm. Second, he stores them in the bank database. Bank generates polynomial function F(x) then creates D(ATM) with it. Bank encrypts D(ATM) with ATM public key and sends it to ATM.

Algorithm1 :registration ATM with BANK

```
ATM establishes communication with Bank
SET bank_pu, bank_pr
SET F(X) = x^a + bx + c
READ $ID_{ATM}$
SEND ATM-certificate to BANK for verification
if failure then Display(“Bad certificate”) and exit
end if
SET $D_{ATM}$=F(ID $_{ATM}$)
SEND E ($D_{ATM}$) to ATM
```

In phases of user registration with the bank, at the first we choose a random number R for each user then generate D(bank,user) that is equal to R. Now, calculate $F_{new}(x)$ which $F_{new}(0)=R$ and choose a random number I for ID user. Bank generates D(user)= $F_{new}(i)$ then encrypt and puts it on a smart card.

Algorithm2 :registration USER with BANK

```
SET user_pu, user_pr
SET random number (R)
SET D(BANK,USER)=(0,R)
SET Fnew(X)=F(X)+R
SET random number (I)
SET Duser=Fnew(I)
SEND E(Duser)
```

In the phases of user to bank authentication, first user authenticates with ATM. User sends password number to ATM with his smartcard. ATM checks if the password is correct. ATM extracts the Duser and Datm.

Algorithm3 authentication between user and ATM

```
Display("Please enter password")
passwd= get input from user()
verifyUsingSC(passwd)
if verified then Authentication Successful
READ Duser,Datm
else Authentication Failed
```

In the user and ATM authentication phase with the bank, at the first, bank retrieves D(bank,user) whit user ID. Then, bank calculates D'(ATM). Finally, bank interpolates new function F'new(x) with three points D'(ATM), D(user), D(bank,user). To verify the authentication among entities bank checks if $F'_{new}(x) == F_{new}(x)$.

Algorithm4 authentication among BANK,ATM,USER

```
Bank read D(bank,user)
SET D'atm=Datm+R
SET interpolation F'new
with(D'atm,Duser,D(bank,user))
IF F'new(x)=Fnew(x) then authentication successful
else
Authentication failed
```

5. PROTOCOL SECURITY ANALYSIS

Security properties required by ATM systems include confidentiality, authentication, integrity and non-repudiation.

Confidentiality

Confidentially means guaranteeing data and important user information not to be accessed by unauthorized users and aliens that usually is performed using cryptography techniques.We encrypt messages such as shares to establish confidentiality in the communication.Thus, dues to the shares that we have considered exclusively for each entity, to access the main secret data, it is required that all the entities must be present, which results in establishing the confidentiality. Also the bank and ATM key pair areused to encrypt messages.This keypair used between the bank and the user prevents the ATM from accessing the information shared between these two entities, which in turn results in the confidentiality. Confidentiality of our algorithm depends on the number of key bits and encryption algorithm. We use RSA as a robust encryption algorithm. There is a trade-off between number of key bits and performance of the algorithm. If the number of key bits is large, protocol security is higher but encryption and decryption process may take longer.

Integrity

Integrity is needed to prevent and discover redundancy, modification, and deletion of data. While registering an ATM and a user in a bank, two independent certificates are issued for these two entities. Both certificates are signed by the user.To establish integrity, we used digital signature in the protocol communication. In SFAMSS, if an attacker tries to change part of the message, receiver can detect the changes by verifying the signature of the message.

Non-Repudiation

Non-repudiation prevents the sender from denying the transmission of her message. We create this mechanism using the signature in the scenario.Since all the signatures in our protocol use private key of the sender, nobody canrepudiatesthe message that he sents before.Because the only entity who has the private key is the sender of a message.

Freshness

In our protocol, we use the timestamp to keep our message freshness. It ensures that the messages are recent, and it ensures that no attacker replayed old messages.Hence, if an attacker uses her previous messages to access the information, the bank detects this situation, due to the existence of the timestamp.

6. CONCLUSION

Our contribution in this paper is to present a framework for ATM systems. Our framework ensures confidentiality and non-repudiation and integrity in communication between bank, ATM, and user. In the registration phase of ATM, bank, and user, secret shares are distributed among users by bank SSS module. In the authentication phase of ATM, bank, and user, shares are aggregated to generate a polynomial function. Bank checks the polynomial function to verify if entities are authenticated. Since we use digital signature and encryption algorithms to enhance the security of ATM communication protocol. We conclude that secret sharing schemes are suitable to use in ATM systems. In the future work, we will analyse the protocol with protocol verifier tools such as prove if [13].

REFERENCES

[1] Z. Ghafari, F. Abazari, M. Analoui, Atmsec: A secure protocol for atm machines, in: Second Computer Science Conference on Computer and Information Technology, 2014.

[2] S. Narani, et al., Social secret sharing for resource management in cloud, arXiv preprint arXiv:1302.1185.

[3] E. D. Canedo, R. De Oliveira Albuquerque, et al., Trust model for reliable file exchange in cloud computing., International Journal of Computer Science & Information Technology 4 (1).

[4] J. L. Dautrich, C. V. Ravishankar, Security limitations of using secret sharing for data outsourcing, in: Data and Applications Security and Privacy XXVI, Springer, 2012, pp. 145-160.

[5] M. A. Hadavi, R. Jalili, Secure data outsourcing based on threshold secret sharing; towards a more practical solution, in: Proc. VLDB PhD Workshop, 2010, pp. 54-59.

[6] Fourar-Laidi, Hakim. "A smart card based framework for securing e-business transactions in distributed systems." Journal of King Saud University-Computer and Information Sciences 25.1 (2013): 1-5.

[7] Robinson, P., N. Cook, and S. Shrivastava. "A Framework for the Rigorous Implementation of Security Protocols."TECHNICAL REPORT SERIES-UNIVERSITY OF NEWCASTLE UPON TYNE COMPUTING SCIENCE 937 (2005).

[8] Malin, Bradley, and Latanya Sweeney. "A secure protocol to distribute unlinkable health data."AMIA Annual Symposium Proceedings.Vol. 2005.American Medical Informatics Association, 2005.

[9] Gorbunov, Serge, and Arnold Rosenbloom. "Autofuzz: Automated network protocol fuzzing framework." IJCSNS 10.8 (2010): 239.

[10] Mittra, Suvo. "Iolus: A framework for scalable secure multicasting." ACM SIGCOMM Computer Communication Review. Vol. 27.No. 4.ACM, 1997.

[11] Kona, Manoj Kumar, and Cheng-ZhongXu. "A framework for network management using mobile agents." Parallel and Distributed Processing Symposium, International. Vol. 2.IEEE Computer Society, 2002.

[12] Camenisch, Jan, Anna Lysyanskaya, and Gregory Neven. "Practical yet universally composable two-server password-authenticated secret sharing."InProceedings of the 2012 ACM conference on Computer and communications security, pp. 525-536.ACM, 2012.

[13] Cheval, Vincent, and Bruno Blanchet. "Proving more observational equivalences with ProVerif."Principles of Security and Trust.Springer Berlin Heidelberg, 2013. 226-246..

[14] B. Schneier, Applied cryptography: protocols, algorithms, and source code in C, john wiley& sons, 2007.

[15] M. K. Alam, et al., An approach secret sharing algorithm in cloud computing security over single to multi clouds. DOI= http://doi.acm.org/10.1145/161468.161471.

[16] Ding, W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report.University of Maryland at College Park.

[17] Fröhlich, B. and Plate, J. 2000. The cubic mouse: a new device for three-dimensional input. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (The Hague, The Netherlands, April 01 - 06, 2000).CHI '00. ACM http://doi.acm.org/10.1145/332040.332491

[18] Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[19] Sannella, M. J. 1994 Constraint Satisfaction and Debugging for Interactive User Interfaces. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington.

[20] Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. 3 (Mar. 2003), 1289-1305.

[21] Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE. In Proceedings of the 16th Annual ACM Symposium on User interface Software and Technology (Vancouver, Canada, November 02 - 05, 2003). UIST '03. ACM Press, New York, NY, 1-10. DOI= http://doi.acm.org/10.1145/964696.964697

[22] Y.T. Yu, M.F. Lau, "A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions", Journal of Systems and Software, 2005, in press.

[23] Spector, A. Z. 1989. Achieving application requirements. In Distributed Systems, S. Mullender, Ed. Acm Press Frontier Series. ACM Press, New York, NY, 19-33. DOI= http://doi.acm.org/10.1145/90417.90738

[24] Chaki, S., &Datta, A. (2009, July). ASPIER: An automated framework for verifying security protocol implementations. In Computer Security Foundations Symposium, 2009.CSF'09. 22nd IEEE (pp. 172-185). IEEE.