Online Scheme Towards Secured Inter Banking Fund Transfer

Aakash Roy 1, **Ajeet Kumar** 2, **Debasish Jana** 3' **Debasis Mitra** 4 and **Radha Tamal Goswami** 5
IT Department, Simplex Infrastructures Ltd, Kolkata

ABSTRACT

On-line fund transfer is becoming an important constituent of today's business need. However, most of the existing frameworks for on-line money transfer require sharing of some private information among involved parties and consequently tempts potentially high risks. Implementation of these schemes also requires significant overhaul of the existing infrastructure. In this paper, we propose a reliable scheme for secured online interoperable fund transfer among banking institutions. In our approach, we facilitate minimal sharing of individual user's private data among banking institutions and eliminate the urge of complete overhaul of any existing framework for its implementation. The proposed scheme is based on loosely coupled service orientation using SOAP and web services with exchange of documents of open standards like XML. The approach has inherent capability of providing authenticity and higher security with protection of confidentiality of information involved in the transfer at an affordable computational time.

KEYWORDS

Electronic Fund Transfer, Web Service, XML, SOAP, Security, Interoperability, SOA.

1. INTRODUCTION

Electronic Fund Transfer among banks for individual consumer or corporate business need is of utmost importance today. Sophistication in software technologies with associated advanced heterogeneous computing architecture and framework demands for seamless integration with interoperability and greater security. Southerd et al [18] presented that customer demands are forcing banks to provide online services. While predicting the future of e-banking, they emphasized that the largest banks would continue to innovate in service bundling with newer ways and adoption of newer technologies. In near future, online banking, as it has started already, as it has started already, would become ubiquitous in nature.

Online electronic fund transfer among two or more banking institutions can only be facilitated if all of them have mutually agreed to a set of protocols. This mutual agreement may involve sharing of some private information or exposure of scripts and methods, which may further involve modification of existing systems belonging to individual banks. In this paper, we attempt to present a scheme for online secured inter banking fund transfer with minimal sharing of private data of individual users among banking institutions. At the same time, we endeavor to eliminate the inevitability of complete overhauling of existing architecture and framework for carrying out its implementation challenges. In the process, we aspire to present one reliable online fund transfer solution for efficient and cost effective inter banking.

In our proposed approach, we have devised a web service to accomplish online fund transfer among banking institutions using the Simple Object Access Protocol (SOAP) architecture with interoperable open standards like extensible Markup Language (XML). The implementation of SOAP in the web services bestows interoperability with increased compatibility across dissimilar architectural infrastructure belonging to the individual banking institutions. Authentication and security are prime contemplation of exposing services as published XML documents over an un-trusted network belonging to public dominion.

The paper is organized as follows. Section two discusses the objective and motivation. Section three follows with background of the relevant scenario with related work done by several other researchers. Our approach towards achievable solution architecture is presented in section four. This section highlights the framework and architecture with interoperability and security support. Section five presents our implementation and technologies used. We conclude with final comments and direction of future work in section six.

2. OBJECTIVE AND MOTIVATION

Online fund Transfer is a very sensitive transaction that may take place over unsecured public medium like the Internet. Any sort of security loophole may compromise the security of the underlying transaction by exposing the transacting data to the public realm. Acts of counterfeit and illegal attacks are extremely likely. Study of existing frameworks reveals that such fund transfers are usually done through third party reliable service providers and the banks (falls back to customers as service levy) have to incur a high cost for the transfer and may take prolonged time for verification and validation. The participating banking organizations in the transfer process also have to agree on a particular set of protocols for availing online fund transfer facilities provided both the parties involved in the transfer must have an account with either of the banks. If a particular bank intends to support compatible fund transfer mechanisms with all other banks, it has to sign up or register with a huge number of third party service providers. This makes the whole process complicated in order to keep track of and maintain. The overall system management thus becomes cluttered with chances of security compromise. Because, the participating bank may have to expose its sensitive confidential data or methods (for remote invocation) to a number of third party organizations. This may also necessitate modifying their existing architecture with complete overhaul, which might be expensive and may incur huge costs. In the light of this context, we need to address this with acceptable security and interoperability needs to provide a platform independent framework for any banking institution to sign up with the process. Also, we need to achieve minimal exposure of transfer information to the outside world, even with the third party reliable consortium and affiliate banks that sign up. Our primary objective is to achieve security at a feasible computational cost. At the same time, we would like to achieve minimal sharing with protection of confidential and sensitive information while communicating over an insecure channel such as the web.

3. BACKGROUND AND RELATED WORK

In today's economy, money (fund, in more generic term) has evolved to electronic or digital form in addition to its physical form as token or paper form. Schoter et al [16] presented an overview of existing systems that implement money in digital forms. They elaborated Internet based payment protocols, net-cash, net-cheque, micro-payments while addressing the security concerns with the necessity of efficiency and speed in digital fund transfers. Providing acceptable form of security with heavy-duty encryption adoption and protecting confidentiality are of paramount necessity. Panurach [14] talked about the emerging digital cash, electronic fund transfers, e-cash and state-of-the-art electronic payment systems as faster and convenient modes of digital fund transfer with increased flexibility. Digital fund transfer estranges paper usage and also delineates chances of bounced cheques. Anonymity and ambiguity still exist and the risks are vested in the hands of the user. Moreover implementations of heavy-duty encryption algorithms are bound to increase computational cost [14]. A proper compromise between security, flexibility and computing time must be reached in order to gain the customer trust with the institution within acceptable time boundaries. Large amount of transfer in electronic form is of major concern with increased risks of potential losses and fraudulence subject to enforcement of law in case of such lapses. Asokan et al [4], Balacheff et al [5] and Diwakar et al [6] have discussed on methods of handling security issues for online fund transfers, in particular, and online transaction, in general. Many other forms of money transfer, both online and offline also exist [13].

Diwakar et al [6] claimed that Indian banking systems follow more social banking with many branches and banking operations are more branch-centric unlike developed countries where centralized banking is prevalent. They presented a cost efficient web centric enterprise level digital information system model that can provide instant access to the entire transaction level data for the bank for decision support and proficient business processes. In another research contribution, Diwakar et al [7] showed that Indian banking system is still passing the nascent stage of Internet banking. The branches of Indian banks need to transform the branches into promotion of business promotion with building of greater customer relationship. Schäfer et al [15] followed a contract-based approach to extend the conventional Web service transaction coordination architecture and infrastructure for supporting flexible compensation operations. Transaction compensation replaces an original invoked but failed operation, by either undoing the results of the original operation or providing similar capabilities as the original one. Liao et al [11] have put emphasis on the foreseeable development of money and banking moving to the cyberspace. In another research work [12], they demonstrated the mounting need of service-quality attributes in Internet banking with absence of face-to-face contacts, as Internet banking differs significantly from traditional brick-and-mortar banking. Alzomai et al [1] presented the urge of development of improved security and integrity of online banking in order to prevent *man-in-the-middle* attacks.

4. PROPOSED APPROACH

With Service Orientation as in Service Oriented Architecture (SOA) [10], in conjunction with web services [8], application integration with cross-platform interoperability, scalability, and availability is attainable. Message oriented communications with loose coupling and asynchronous connection are possible in SOA. Anand et al [3] have preached for about service-based model in banking transactions. They emphasized that in a competitive business environment, a comprehensive SOA strategy is required a business process driven banking domain to provide flexibility, open standards and loose coupling. Our framework and architecture gets the motivation from these research works by several researchers and we have implemented our architecture testbed using existing technologies and platforms. Our strong motivation is from the formal theoretical model, presented by Gordon et al [9] who used an abstraction for web services using SOAP-level security. They formally showed that no vulnerability exists to attacks representable within the spi-calculus [2] within the premises of certain assumptions. They further emphasized that vulnerabilities may exist outside their model, however, there are no methods, formal or otherwise, to guarantee absolute security [9]. Tang et al [20] and Thomson [21] showed the use of web services as loosely coupled interoperable mode of service orientated communication backbone. We adopt this affordable technology for interoperability among diverse platforms of the participating organizations and providing high level of security over the public domain like web transactions.

4.1 THE FRAMEWORK AND ARCHITETCURE

While presenting our secured interoperable online fund transfer scheme based on available and affordable technologies, we maintain the most important and prevalent ACID properties [16]. We present a consortium as the intermediary of all participating banks providing the layer of message exchanges over secure protocols in insecure medium like web. A simple architectural diagram depicting an abstract view of the consortium is shown in Figure 1.

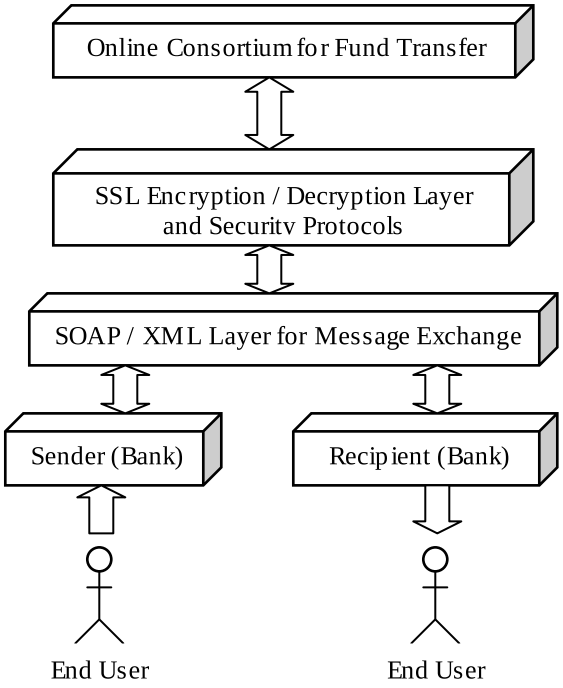
Online Consortium for Fund Transfer
SSL Encryption / Decryption Layer and Securitv Protocols
SOAP / XML Layer for Message Exchange
Sender (Bank)
Recipient (Bank)
End User
End User

Figure 1: Architectural diagram of the consortium

At the lowest layer, we have the end users who shall be interacting with their respective banking organizations. When a user requests for money transfer to another account located at a different bank, the sender bank processes the request first in order to check and see if the user has sufficient funds to send. After all necessary checking, the sender bank establishes a secure session with the consortium. This involves a handshaking process, where both the bank and consortium authenticate each other to carry out the transfer. Now, the sender bank creates an XML message that contains encrypted transfer information embedded within the appropriate tags. This encryption is carried out with public key cryptography with digital signatures to ensure both authentication and security [19]. This XML message is encapsulated within SOAP envelope and then this envelope is further encrypted at the Secured Sockets Layer [13] at a more granular level. This information is sent to the consortium. The consortium decrypts the information and then performs some housekeeping. These include validating the XML data using schemas, entering relevant transfer information into its database for tracking the transaction status and maintaining a legal proof of transfer.

The consortium then establishes a session with the receiver bank, which involves a handshaking phase for authenticating each other. Once the session is established, the consortium creates a fresh XML document containing transfer details obtained from the sending bank. This is done to further re-establish the genuine-ness of the transaction and that it is now a transaction to be carried out by the consortium. The information in the XML document is encrypted within the appropriate XML tags and this document is then encapsulated within a SOAP envelope and is sent to the server of the receiving bank. The receiving bank decrypts the information and then updates the database. It must also inform the consortium about a successful transfer; else the status of the transfer shall remain pending with the consortium. Only after conformation from the receiver bank shall the consortium send a status complete to the sender bank, which will then debit the amount from the sender's account. The above-mentioned methodology implicitly implies some norms (for example, using XML as the standard for information interchange, adhering to the security features, etc.), which the banking organizations must adhere to. Let us illustrate the proposed scheme by considering a sample scenario. Let us assume that two banks B1 and B2 are registered under the consortium and want to initiate a transfer. All transfer requests and information exchange are carried out using XML messages within a SOAP envelope and through the SSL (Secure Sockets Layer), which inherently provides encryption/decryption for exchange of data. The XML messages contain encrypted information that uses public key cryptography and is digitally signed.

4.2 INTEROPERABILITY

XML provides the medium of interoperability. SOAP with web services perfectly fit in to support service orientation across platforms and diverse technologies. Being interoperable, and platform independent, XML messages carry no platform specific information. As such, the scheme perfectly suits the need of inter banking transfer of messages using secured protocols described separately. An example of XML message scheme for sending request to consortium for a fund transfer is given in Figure 2.

Before the Internet and the Web came into existence, communication between systems was very tightly coupled with the underlying system. Today's complex software needs to be interoperable and loosely coupled in order to communicate with each other in a seamless manner. Earlier SOAs were built using DCOM or Object Request Brokers (ORBs) based on the Common Object Request Broker Architecture (CORBA) specification. CORBA was capable of providing standards-based interoperability and connectivity. SOAs provide loose coupling between components so that some components provide services while some others consume offered services. Web services promise to offer interoperability to cater to the growing complexity needs of software yet at the same time provide easy to access service interfaces [10].

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:EnvelopeSOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/en
coding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="urn:moneyTransferWSDL">
<SOAP-ENV:Body>
<tns:moneyTransferRequest
xmlns:tns="urn:moneyTransferWSDL">
<requestingBank
xsi:type="xsd:string">EJYO/Y…IO</toBankid>
<requestedBank
xsi:type="xsd:string">FFGT//YI..OUYT</fromBankid>
</tns:moneyTransferRequest></SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figure 2: XML Schema for sending request to consortium for transfer

At the consortium, the information is decrypted by using the public key of the bank and the private key of the consortium. The consortium thereby validates the requesting bank with secure means. It now checks for availability of the bank to which the transfer is to be made. The consortium sends a session token and a md5 hash of the timestamp to the requested bank by first encrypting this information by its private key and then by the public key of the requested bank. The md5 hash of the timestamp forms the random shared secret key. The *sessiondetails* tag represents this information. The requested bank decrypts the information and then responds with its own identifier and the session token. Since this information is encrypted using public key signatures, the consortium, after decrypting the message, knows the authenticity of the requested bank.

Next, the consortium sends the information to the requesting bank as a response for initiating the transfer. Thus, at each point of time in communication, the appropriate public key encryption and signatures are being used, specific to a particular bank. The stage is now set for the transfer and a valid session has been established.

4.3 SECURITY

The proposed security and authentication scheme has been derived from the use of public key signatures [19]. Public key signatures involve the use of both private and public key pairs for encryption and decryption purposes, thus assuring authentication as well as secrecy. Security comes with some additional cost but the benefits are worth adopting the increased cost. We propose a mechanism whereby the computational cost can be reduced by the use of a random shared secret key encryption mechanism. Thus, our proposed mechanism operates in two phases. One, in which authentication is verified and then, secrecy of transfer is achieved. While initiating a transfer, once the consortium has verified the authenticity of the banks that wish to participate in the transfer, all subsequent transfer messages that are exchanged will be encrypted using a shared secret key that would only be valid and unique for a particular transaction. All transactions are time stamped to prevent replay attacks and attacks of similar nature.

It is worthwhile to mention here that in order to achieve end-to-end security, relying on the HTTPS/SSL transport security features would not be exhaustive. Although it is required as a secured channel of transfer, this would not suffice to the extent where the requirement for a message level security is paramount. We now present a sample scenario that emphasizes on the security aspects of the system. Figure 3 shows the sequence diagram of the scenario. Bank A wants to transfer money from an account that it maintains to an account maintained at bank B. This is how the authentication and security scheme may be implemented:

1. Bank A sends to the consortium, its identification and the bank to which it wants to transfer money, encrypted with its private key and then with the public key of the consortium.
2. The consortium decrypts the contents, first with its private key and then with the public key of bank A.

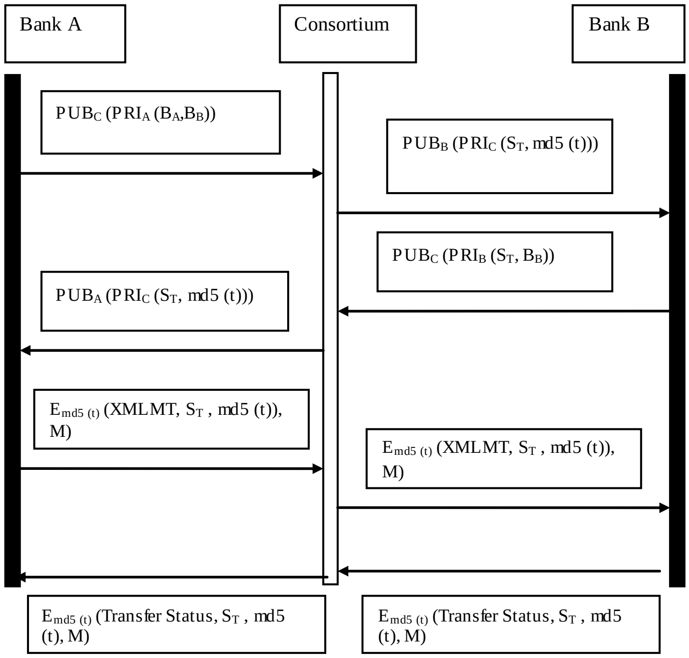
Bank A
Consortium
Bank B
$PUB_C (PRI_A (B_A,B_B))$
$PUB_B (PRI_C (S_T, md5 (t)))$
$PUB_C (PRI_B (S_T, B_B))$
$PUB_A (PRI_C (S_T, md5 (t)))$
$E_{md5 (t)} (XMLMT, S_T , md5 (t)), M)$
$E_{md5 (t)} (XMLMT, S_T , md5 (t)), M)$
$E_{md5 (t)} (Transfer Status, S_T , md5 (t), M)$
$E_{md5 (t)} (Transfer Status, S_T , md5 (t), M)$

Figure 3 : A sequence diagram of a sample scenario

3. The consortium now knows that bank A wants to transfer money to bank B. It generates a session token and an md5 hash of the current timestamp, encrypts the two using its private key and bank B's public key, and then sends it to bank B
4. Bank B decrypts the contents and indicates that it is ready to accept the transfer by sending back the session token and its identification, encrypted by its private key and the public key of the consortium
5. The consortium sends a request to bank A indicating it to send in the transfer details. It sends bank A the session token and the md5 hash of the timestamp by encrypting the

information with its private key and the public key of bank A.

6. At this stage, the authentication of all parties involved in the transfer has been verified by the consortium. A valid session has been established for the transfer to take place. The md5 hash of the timestamp becomes the shared secret key for exchange of transfer information. The session token is now used as a reference of an agreement that all further information exchanges for this particular transfer are genuine and authentic.
7. Bank A now sends the encrypted XML document that states the details of the transfer, along with the session token and the md5 hash of the timestamp, to the consortium using the shared secret key.
8. The consortium decrypts the information in the XML document, validates the timestamp and the session, performs some housekeeping activity and sends the encrypted XML document along with the session token and the md5 hash of the timestamp to bank B. It encrypts this message using the shared secret key.
9. Bank B decrypts the details and ensures that the session and timestamp is valid, and performs the necessary updates in its data repository(s).

Legends used in Figure 3:
A (in subscript): Bank A
B (in subscript): Bank B
C (in subscript): Consortium
PUB_i: The Public key of the respective organization, i
$PRI_{i:}$ The private key of the respective organization, i
S_T: Session token, generated by the consortium
$E_{md5\ (t)}$: Encrypted messages using a shared secret key, md5 (t)
M: Message digest, the md5 hash of the actual transfer message
md5 (t): The md5 hash of the timestamp of the start of transfer, generated by the consortium
XMLMT: The XML for money transfer over SOAP

It is worthy to mention here that the information on the XML document should not be exposed. If we use public key signatures for both security and authentication, then it would be computationally expensive to achieve a decent pace at which the transfer can take place. Thus, we have utilized the benefits of public key signatures to authenticate the parties involved in the transfer and communicate the shared secret key. This ensures that at any stage, no information is revealed to the insecure channel and at the same time it proves to be less expensive computationally. Moreover, since transfer details may be lengthy, it might not be a wise idea to implement public key signatures and two way encryptions for the transfer to take place as far as the overall computational cost is concerned

5. IMPLEMENTATION AND TECHNOLOGIES

We have used the following technologies: PHP (version: 5.2.6) (Server side scripting), XHTML/JAVASCRIPT (Client side scripting), MySQL (version: 5.0.51b) (Database), XML Specification (W3C Standard), the SOAP (version 1.2) Specification (W3C Standard), NuSoap, a PHP library consisting of reusable classes to implement the SOAP transfer and requests via XML. We have implemented the proposed architecture using PHP and MySQL on an Apache (version: 2.2.8) Web Server. However the security features are still under development and are yet to be implemented. We used Apache on a windows platform (WAMPSERVER version: 2.0c [22]). The testbed implementation of the banks and the consortium has been achieved by using various scripts programmed in PHP that represent the necessary servers.

In order to implement the security features in the system, an industry standard is required since the service has to be platform independent. Since our system would finally model a web service, we have implemented the security features using the Web Service security standards. SOAP security extensions have also been used. SQL anti-injection schemes have been used so that SQL injection attacks cannot be performed in the system by the use of queries provided at the user interface. Since the consortium was implemented using PHP, we used a built in SQL anti-injection mechanism feature.

6. CONCLUSION AND FUTURE WORK

Security has been implemented in a way that provides both authenticity and confidentiality of information involved in the transfer at an affordable computational time. This adds to the novelty of the system over existing ones. The solution is also cost effective. Some of the key objectives of the system have been achieved including Model Money Transfer Solution for all banks with easy yet robust registration process, extensible, as transfer is based on XML using SOAP, platform independent. Thus, no existing change of infrastructure is required for any bank that wishes to use the service. There is no need to expose any bank's personal/private data over the Internet. Transfer details are kept as highly abstract as possible and are also encrypted. We achieve portability and quick Transfer with immediate transaction status and tracking. XML can be validated using Schema and DTD for a Valid/Invalid Transaction format. Additional layers of security for authentication and secrecy of transferred information.

Our primary motive and agenda was to implement SOAP as a standard for data and information interchange for effective inter-banking and we were successful in our attempts. The beauty of implementing SOAP lies in the fact that it provides cross platform compatibility across all systems and is highly portable and scalable. Thus we do not need to reform any existing structure to implement our system. Our approach has inherent capability of providing authenticity and higher security with protection of confidentiality of information involved in the transfer at an affordable computational time.

The existing system can be extended to offer more inter-banking facilities. Moreover, if this system can be integrated with online payment systems, users will be able to check out directly using their bank account information. A challenge, however, would be to make such a system more secure against threats and attacks. The successful implementation of this system is after all a matter of mutual trust among banking institutions who wish to participate. Such a system can only come into existence if appropriate laws enforce its principles.

REFERENCES

[1] AlZomai, M., AlFayyadh, B., Jøsang, A., and McCullagh, A. 2008. An exprimental investigation of the usability of transaction authorization in online bank security systems. In *Proceedings of the Sixth Australasian Conference on information Security - Volume 81* (Wollongong, NSW, Australia, January 01 - 01, 2008). L. Brankovic and M. Miller, Eds. ACM International Conference Proceeding Series, vol. 328. Australian Computer Society, Darlinghurst, Australia, pp. 65-73.

[2] Abadi M. and Gordon A.D.., A calculus for cryptographic protocols: The spi calculus. *Information and Computation*, 1999, 148:1–70.

[3] Anand, S. and Venkataraman, J. 2006. Drivers for SOA in the Transaction Banking Domain. In *Proceedings of the IEEE international Conference on Services Computing* (September 18 - 22, 2006). IEEE Computer Society, Washington, DC, 506-507.

[4] Asokan, N., Janson, P. A., Steiner, M., and Waidner, M. 1997. The State of the Art in Electronic Payment Systems. *Computer* 30, 9 (Sep. 1997), 28-35.

[5] Balacheff, B., Chen, L., Plaquin, D., and Proudler, G. 2001. A trusted process to digitally sign a document. In *Proceedings of the 2001 Workshop on New Security Paradigms* (Cloudcroft, New Mexico, September 10 - 13, 2001). NSPW '01. ACM, New York, NY, 79-86.

[6] Diwakar, H. and Naik, A. 2006. Enterprise Wide Web Centric Digital Information System for Indian Banks. In *Proceedings of the First International Conference on Digital Information Management, 2006* (Dec 06 - 06, 2006). 89-95.

[7] Diwakar, H. and Vaidya, A. 2008. Information Analysis for Enabling Optimal Utilization of Service Delivery Channels in Indian Banks. In *Proceedings of the Seventh IEEE/ACIS International Conference on Computer and information Science (Icis 2008) - Volume 00* (May 14 - 16, 2008). International Conference on Information Systems. IEEE Computer Society, Washington, DC, 146-151.

[8] Fujita Satoru, Dynamic Collaboration of Businesses using Web Services, *NEC Journal of Advanced Technology*, Vol. 1, No. 1, Jan, 2004, P. 36-42.

[9] Gordon, A. D. and Pucella, R. 2002. Validating a Web service security abstraction by typing. In *Proceedings of the 2002 ACM Workshop on XML Security* (Fairfax, VA, November 22 - 22, 2002). XMLSEC '02. ACM, New York, NY, 18-29.

[10] Jana D. 2006. “Service Oriented Architectures – A New Paradigm”, *CSI Communications*, Computer Society of India, March 2006,12-14.

[11] Liao, Z. and Cheung, M. T. 2003. Challenges to Internet e-banking. *Commun. ACM* 46, 12 (Dec. 2003), 248-250.

[12] Liao, Z. and Cheung, M. T. 2008. Measuring consumer satisfaction in internet banking: a core framework. *Commun. ACM* 51, 4 (Apr. 2008), 47-51.

[13] Money Transfer, Wikipedia Notes, http://en.wikipedia.org/wiki/Money_transfer

[14] Panurach, P. 1996. Money in electronic commerce: digital cash, electronic fund transfer, and Ecash. *Commun. ACM* 39, 6 (Jun. 1996), 45-50.

[15] Schäfer, M., Dolog, P., and Nejdl, W. 2008. An environment for flexible advanced compensations of Web service transactions. *ACM Trans. Web* 2, 2 (Apr. 2008), 1-36.

[16] Schöter A. and Rachel W., “Digital Money Online - A Review of Some Existing Technologies”, *Article from http://www.intertrader.com*, Feb. 1997.

[17] SOAP Version 1.2: Messaging Framework (Second Edition), *W3C Recommendation,* 27 April 2007, http://www.w3.org/XML

[18] Southard, P. B. and Siau, K. 2004. A survey of online e-banking retail initiatives. *Commun. ACM* 47, 10 (Oct. 2004), 99-102.

[19] Tanenbaum A. S, *Computer Networks*, ISBN: 978-81-203-2175-5, PHI Learning Private Limited, Fourth Edition.

[20] Tang, K., Chen, S., Levy, D., Zic, J., and Yan, B. 2006. A Performance Evaluation of Web Services Security. In *Proceedings of the 10th IEEE international Enterprise Distributed Object Computing Conference* (October 16 - 20, 2006). EDOC. IEEE Computer Society, Washington, DC, 67-74.

[21] Thomson, S., Implementing WS-Security A case study, April 2003, http://www.ibm.com/developerworks/webservices/library/ws-security.html

[22] WamServer Notes, “Apache, MySQL, PHP on Windows”, www.wampserver.com/en

[23] XML Specifications, *W3C Standards*, Modified October, 2008, http://www.w3.org/XML