Journal of Internet Banking and Commerce

An open access Internet journal (http://www.arraydev.com/commerce/jibc/)

Journal of Internet Banking and Commerce, April 2009, vol. 14, no.1
(http://www.arraydev.com/commerce/jibc/)

A Secured Hybrid Architecture Model
for Internet Banking (e-Banking)

Ganesan R
Senior Lecturer in Department of Computer Science
Postal Address: PSG College of Arts and Science, Coimbatore, TamilNadu, India – 641 014
Email: rganrao@gmail.com

Vivekanandan K, Ph.D.
Reader, BSMED
Postal Address: Bharathiar University, Coimbatore, Tamilnadu, India

Abstract

Internet banking has made it easy to carry out the personal or business financial transaction without going to bank and at any suitable time. This facility enables to transfer money to other accounts and checking current balance alongside the status of any financial transaction made in the account. However, in order to maintain privacy and to avoid any misuse of transactions, it is necessary to follow a secured architecture model which ensures the privacy and integrity of the transactions and provides confidence on internet banking is stable. In this research paper, a secured hybrid architecture model for the internet banking using Hyperelliptic curve cryptosystem and MD5 is described. This hybrid model is implemented with the hyperelliptic curve cryptosystem and it performs the encryption and decryption processes in an efficient way merely with an 80-bit key size. The various screen shots given in this contribution shows that the hybrid model which encompasses HECC and MD5 can be considered in the internet banking environment to enrich the privacy and integrity of the sensitive data transmitted between the clients and the application server.

Keywords: **E-Banking, Hyperelliptic curve cryptosystem, MD5, Authentication, Confidentiality, Integrity, Nonrepudiation, Privacy,**

INTRODUCTION

e-Banking is defined as the automated delivery of new and traditional banking products and services directly to customers through electronic, interactive communication channels. e-Banking includes the systems that enable financial institution customers, individuals or corporates to access accounts, transact business, or obtain information on financial products and services through a public or private network, like internet or mobile phone. Internet banking (also referred as e-banking) is changing the banking industry and is having the major effects on banking relationships. Banking is now no longer confined to the branches where one has to approach the branch in person to withdraw cash or deposit a cheque or request a statement of accounts. In true Internet banking, any inquiry or transaction is processed online without any reference to the branch (anywhere banking) at any time. Providing Internet banking is increasingly becoming a "need to have" than a "nice to have" service. The net banking, thus, now is more of a norm rather than an exception in many developed countries due to the fact that it is the cheapest way of providing banking services.

The main issue of the internet banking analyzed during the survey conducted by online banking association in the year 2002 was the matter of security. Security is a crucial requirement of an e-commerce system due to the fact that the sensitive financial information that these systems transmit travel over untrusted networks where it is essentially fair game for anyone with local or even remote access to any part of the path followed. The security is required for dual purposes. They are, i) to protect customers' privacy ii) to protect against fraud.

For the duration, the optimal development strategy for e-banks is likely to be the cultivation of the demand side along the paths of least resistance, in particular as regards consumer perceptions of transaction security, transaction accuracy, user friendliness, and network speed (Liao Ziqi, Cheung M, (2002)). Any Internet banking system must solve the issues of authentication, confidentiality, integrity, and nonrepudiation, which means it, must ensure that only qualified people can access an Internet banking account. The information/transaction thus viewed remains private and cannot be modified, traced or verified by third parties. Most of the attacks on online banking used today are based on deceiving the user to steal login data and valid transaction number (TAN). Two well known examples for those attacks are phishing and pharming. Various security issues of the internet banking are discussed in (Barker E, Barker W, Burr W, Polk W, Smid M, (2007)) (Osama D, Phu Dung Le, Srinivasan B, (2007)) (Barclays Bank (2006)).

In the empirically most important areas of transaction security and accuracy, encryption protocols such as the Secure Sockets Layer (SSL) and Secure Electronic Transactions (SET) have been widely adopted by e-banks (Liao Ziqi, Cheung M, (2003)). Most of the security protocol uses algorithms like IDEA with a 128-bit key or RSA with a 1024-bit key to encode the transaction data (Seitz J, Stickel E, (1998)). But, using HECC with 80-bit key length, one can achieve the same level of security instead of using RSA with 1024-bit key (Barker E, Barker W, Burr W, Polk W, Smid M, (2007)). This proposed hybrid architecture model consist of hyperelliptic curve cryptosystem over prime field F_p of genus 2 and MD5 to resolve the various security issues faced by the internet banking (e-Banking). The overview of the hyperelliptic curve cryptography over prime field is discussed in Ganesan R, Gobi M, Dr. Vivekanandan K (2008). Details of the hyperelliptic curve cryptosystem over prime field F_p of genus 2 and its implementation can be had from Ganesan R, Dr. Vivekanandan, K (2008). Also, the step by step algorithm of MD5 can be had from Dr. Janakiraman VS, Ganesan R, Gobi M (2007).

SECURED HYBRID ARCHITECTURE MODEL FOR E-BANKING

Encryption techniques used by the bank (including the public-key encryption) should ensure that the privacy of data flowing between the browser and the server system is protected. The message digest technique used by the bank should guarantee the integrity of data moving between the bank customer and the server. Following proposed hybrid architecture model (Figure 1) which comprises the hyperelliptic curve cryptosystem over finite field F_p of genus 2 and the MD5 technique, overcomes the security issues called privacy of sensitive data and the integrity of the same data flowing between client and the server.

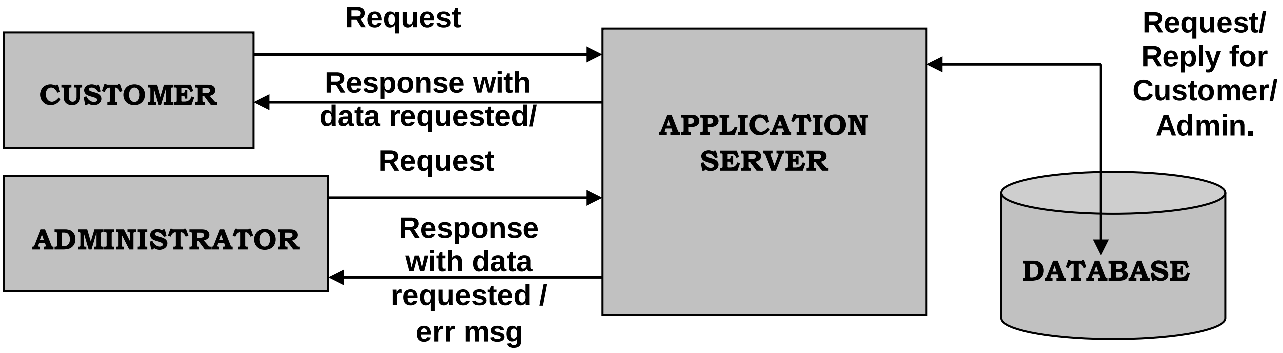

Request
CUSTOMER
Response with data requested/
Request
ADMINISTRATOR
Response with data requested / err msg
APPLICATION SERVER
Request/ Reply for Customer/ Admin.
DATABASE

Figure 1: Secured hybrid architecture model for e-Banking which encompass HECC & MD5

The hybrid architecture shown above has the following three components:

1. **Client:** There are two clients for the application. One is a web-based user-friendly client called bank customers. The other is for administration purposes. Client's / Administrator's request is sent over the network in an encrypted data format. Also, to ensure integrity of the request that is transmitted, it is subjected to hash algorithm. In addition to that, the client / administrator have to decrypt the response data sent by the application server which is in the encrypted format and also verifies the integrity of the received data. The above encryption and decryption process is done using hyperelliptic curve cryptographic technique and the integrity of the data is ensured using MD5 hash algorithm.
2. **Application Server:** It takes care of the server application, JDBC-ODBC driver, and tests for the ODBC connectivity for mapping the database in order to fulfill client's and administrator's request. HECC system in the server decrypts the client's /

administrator's request and verifies the integrity of the request and finally it communicates with the database to perform the request. Subsequently, the reply from the database is encrypted as well as it is subjected to MD5 to ensure integrity and is sent back to the client / administrator.

3. **Database:** Database Server will store customer's details and bank data.

PROCESS FLOW DIAGRAM

This section highlights only the main process flow diagrams for encryption and decryption of a request using hyperelliptic curve cryptosystem with the integrity of the request using MD5 hash algorithm. The figure 2 depicts the encryption of the request and also illustrates how the integrity of the request is made. The request considered here is a bank customer's transaction detail.

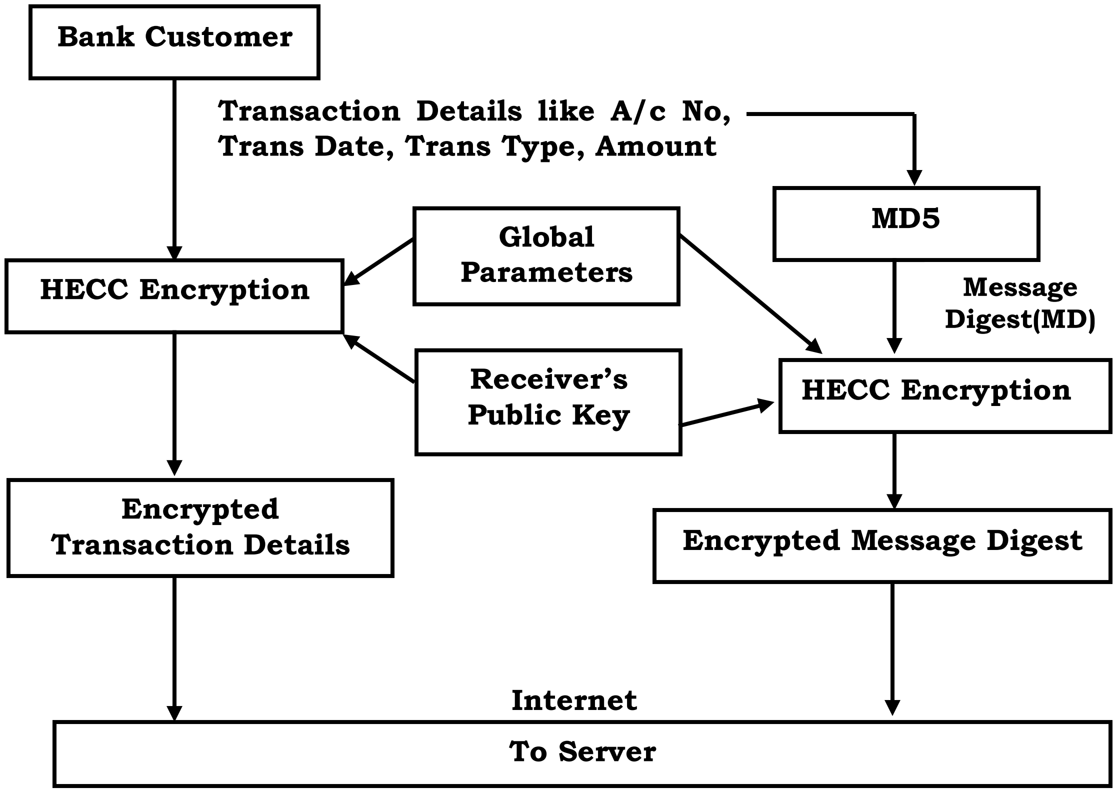


Figure 2: Transaction encryption process flow diagram

In the above process flow diagram, the bank customer's request is transferred securely over the insecure communication channel like internet using hyperelliptic curve cryptosystem and MD5. The global parameters needed are hyperelliptic curve, prime and the divisor. The encryption is done with the help of the receiver's (bank's) public-key.

The message digest of the request (MD) is created using MD5 algorithm. After the transaction details and the message digest are encrypted, the encrypted data is transferred to the server through the insecure e-commerce channel for further processing. The same encryption and integrity processes are performed on the administrator's request. Moreover, the sensitive response data from the server is also encrypted by the application server using client's public-key before transmitting it to the client.

The sequence of operations performed on decryption and integrity verification processes are described in the figure 3. The application server decrypts the encrypted transaction details and the encrypted message digest using hyperelliptic curve cryptosystem with the help of the bank's private key. After the transaction is successfully decrypted, it computes the message digest (md) for the same. Now the application server compares the computed message digest (md) and the decrypted message digest (MD). If both are equal, it stores the original transaction details in the database and acknowledges the client or else rejects the transaction and sends an error message to the client.

The above same processes are performed on sensitive response data by the client using his private key and the MD5.

DATABASE DESIGN

Databases are designed in such a way that it mainly focuses on the hyperelliptic curve global parameters, private and pubic keys, bank customer's details and their transaction details. The structure of the database is as follows:

- *Global_Parameters* table is used to store the global parameters which are used at the time of keys generation, encryption and decryption. Global_Parameters contains prime, hecurve and divisor.

  Structure Definition:

prime	: Prime number generated
hecurve	: Hyperelliptic curve generated
divisor	: Divisor of the hyperelliptic curve

- *User_keys* table is used to store the bank customer's private key and public key. This table contains user_id, private and public.

  Structure Definition

user_id	: Identification of the bank customer.
private	: Private key generated for the customer
public	: Public key generated for the customer

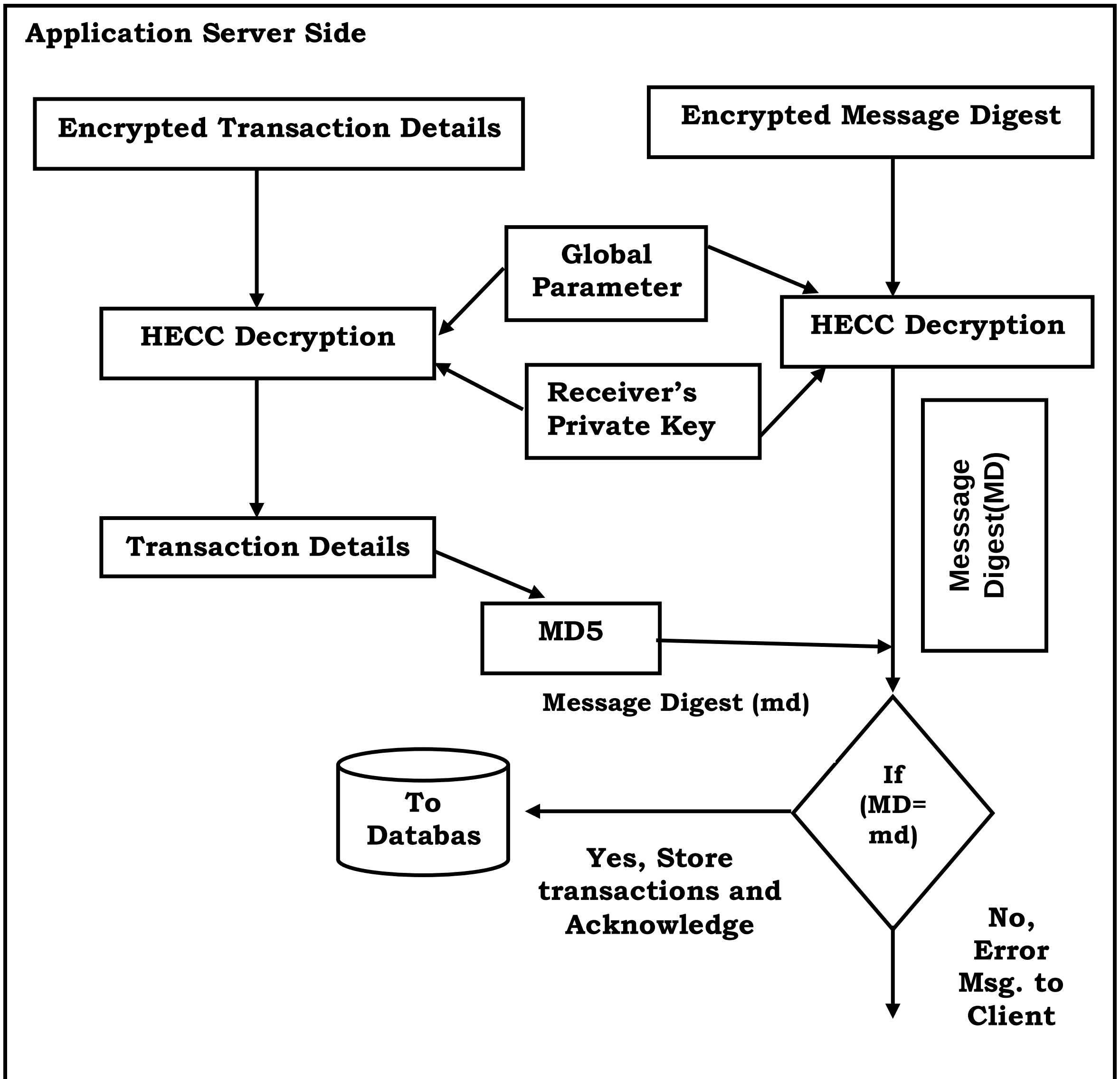


Figure 3: Transaction Decryption process flow diagram

- *Customer_Details* table is used to store the details of the bank customer. This table contains the main details like user_id, pwd, account_no, customer_name, address, account_type, and branch.

Structure Defintion:

user_id	: Identification of the customer
pwd	: Password
account_no	: Account Number of the Bank customer
customer_name	: Name of the account holder

address	: Address of the bank customer
account_type	: Type of the account (Savings or Current)
branch	: Bank branch details

- *Trans_Details* table is used to store the actual transactions performed by the bank customers. This table contains trans_id, account_no, user_id, trans_date, trans_type and amount.

Structure Definiton:

trans_id	: Identification of the customer's transaction
account_no	: Account Number of bank customer
user_id	: Identification of the customer
trans_date	: Transaction Date
trans_type	: Transaction Type (Deposit or Withdrawal)
amount	: Transaction Amount (Rs.)

IMPLEMENTATION DETAILS

The entire hyperelliptic curve cryptosystem is developed using Java server pages (JSP) and the server used is Apache Tomcat Server 6.0. The Back-end used for storing the details is Microsoft SQL Server. The application is executed in Pentium III Celeron processor @ 1GHz speed with 256 MB RAM.

Following are the main methods developed and implemented to transfer sensitive request/response data in the secure e-Banking system.

- User/Administrator Authentication
- Message Digest Creation / Verification
- Hyperelliptic curve generation and divisor generation
- Key (Private key and Public key) generation
- HEC_Encryption
- HEC_Decryption

- *User/Administrator Authentication:*

This specific method is used to authenticate whether the logged in client/administrator is the right person or not. While logging in, the client/administrator enters their user identification and password. The entered data are encrypted using bank's public key and is sent to the application server for verification. After receiving the encrypted data, the application server decrypts it with bank's private key and retrieves the original data. Finally, server compares this user identification and password with the corresponding user identification and password in the database. If both are identical, the application server allows the user to enter into the next screen, otherwise displays an error message.

- *Message Digest Creation / Verification*

MD5 algorithm is mainly used to create the message digest for the request. The purpose of this algorithm is to check the integrity of the message to be transmitted. Java has built-in class and methods to generate the message digest which is described in the java package javax.crypto.* and java.security.*. The size of the message digest is 128-bit. Message digest verification is the process which is done at the receiver's side to validate or compare the two message digests. One is computed at the receiver's side and the other one is transmitted from the sender's side. If both are equal, the integrity of the transmitted message is passed otherwise, it is failed.

- *Hyperelliptic curve generation and divisor generation:*

  This method is mainly used to generate the hyperelliptic curve over prime finite field F_p of genus 2. The input to this method is the length of prime to generate and the coefficients of the hyperelliptic curve. Once the curve is generated, the divisor is generated next. The generated hyperelliptic curve, prime number and the divisor of the curve are stored in the *Global_Parameters* for further reference. Only one hyperelliptic curve over prime field F_p of genus 2 is maintained by the bank for the entire e-Banking application. The administrator executes this method to generate prime, curve and the divisor.
- *Key (Private key and Public key) generation:*

  This method is mainly implemented for generating user (bank customer) keys. There are two keys generated which are referred to as the private key and the public key. Each customer receives one private key and one public key and the same is stored in the *User_Keys.* Private key is kept secret and the public key is known to everyone. Private and public keys are generated and maintained for the bank also. This method is executed by the bank's administrator to generate the keys.
- *HEC_Encryption:*

  The purpose of this method is to protect the sensitive details for the transactions. The sensitive details may be either bank's customer details or their transaction details. For example, once the customer logs in successfully, they enter their sensitive data. Then, the entered data and its message digest are encrypted using bank's public key and is sent to the server. The same encryption method is used by the application server to encrypt the sensitive response data and the message digest.
- *HEC_Decryption:*

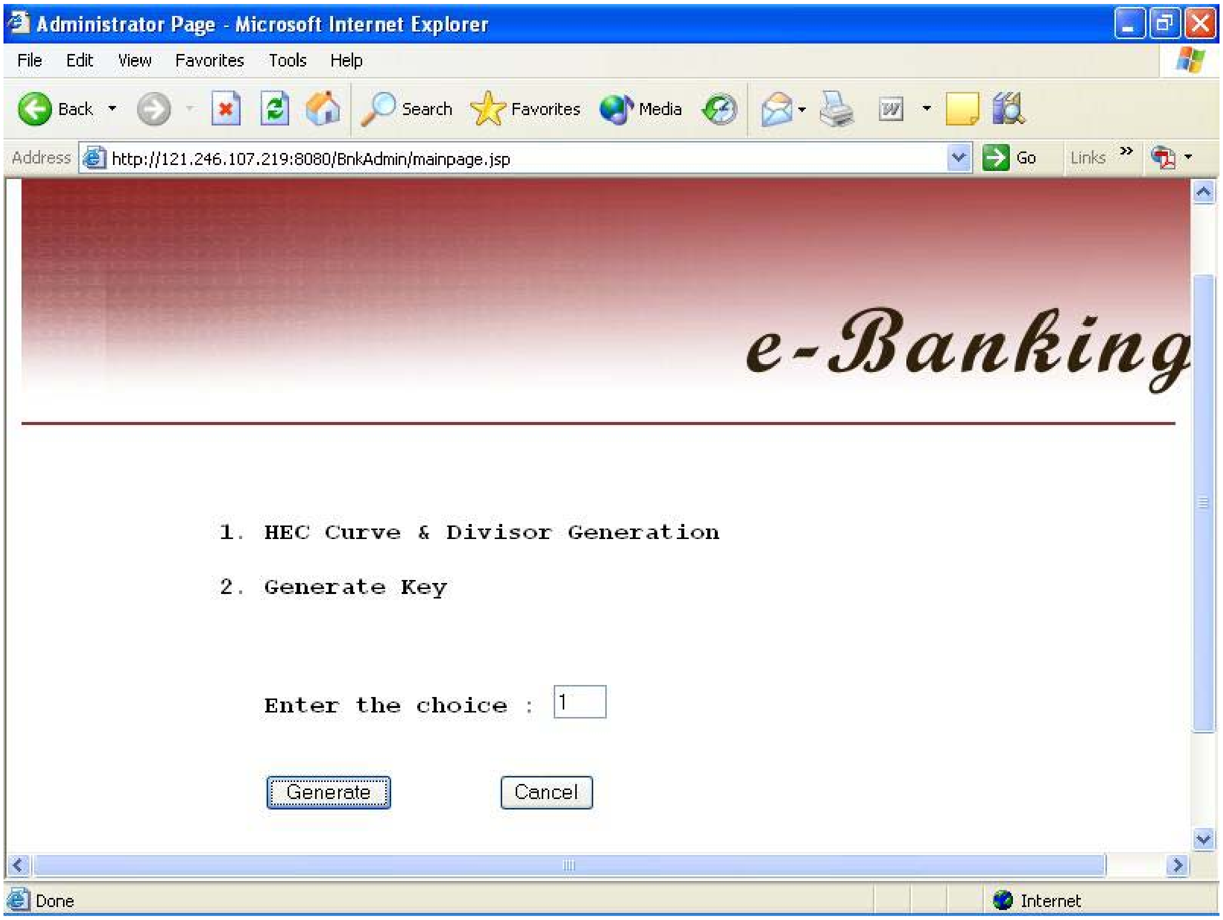
  The main function of this method is to decrypt the encrypted data and the message digest. The application server reads both the encrypted data and the message digest and decrypts the same using bank's private key. The same decryption method is also used by the client to decrypt the sensitive response details which are sent by the application server based on the request made by the client.

CONCLUSION

Information about financial institutions, their customers, and their transactions are, by necessity, extremely sensitive; thus, doing business via a public network introduces new challenges for security and trustworthiness. Given the open nature of the Internet, transaction security is likely to emerge as the biggest concern among the e-bank's account holders. For securely and privately transmitting the data over the Internet, most protocol use both public key and secret key cryptography. To implement public key cryptography, RSA algorithm is used with the key size of 1024-bit. But the above hybrid architecture model is implemented with the hyperelliptic curve cryptosystem and it performs the encryption and decryption processes in an efficient way merely with an 80-bit key size. The main objective of this model is to consider and include the hyperelliptic curve cryptosystem and MD5 in the internet banking environment to enrich the privacy and integrity of the sensitive data transmitted between the clients and the application server.

SAMPLE SCREEN SHOTS

The main screen shots for the secured e-Banking application which is developed using HECC and MD5 are shown below.


Administrator Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://121.246.107.219:8080/BnkAdmin/mainpage.jsp Go Links
e-Banking
Generate
Cancel
Done
Internet

1. HEC Curve & Divisor Generation

2. Generate Key

Enter the choice : 1

Figure 4: Menu Option Screen

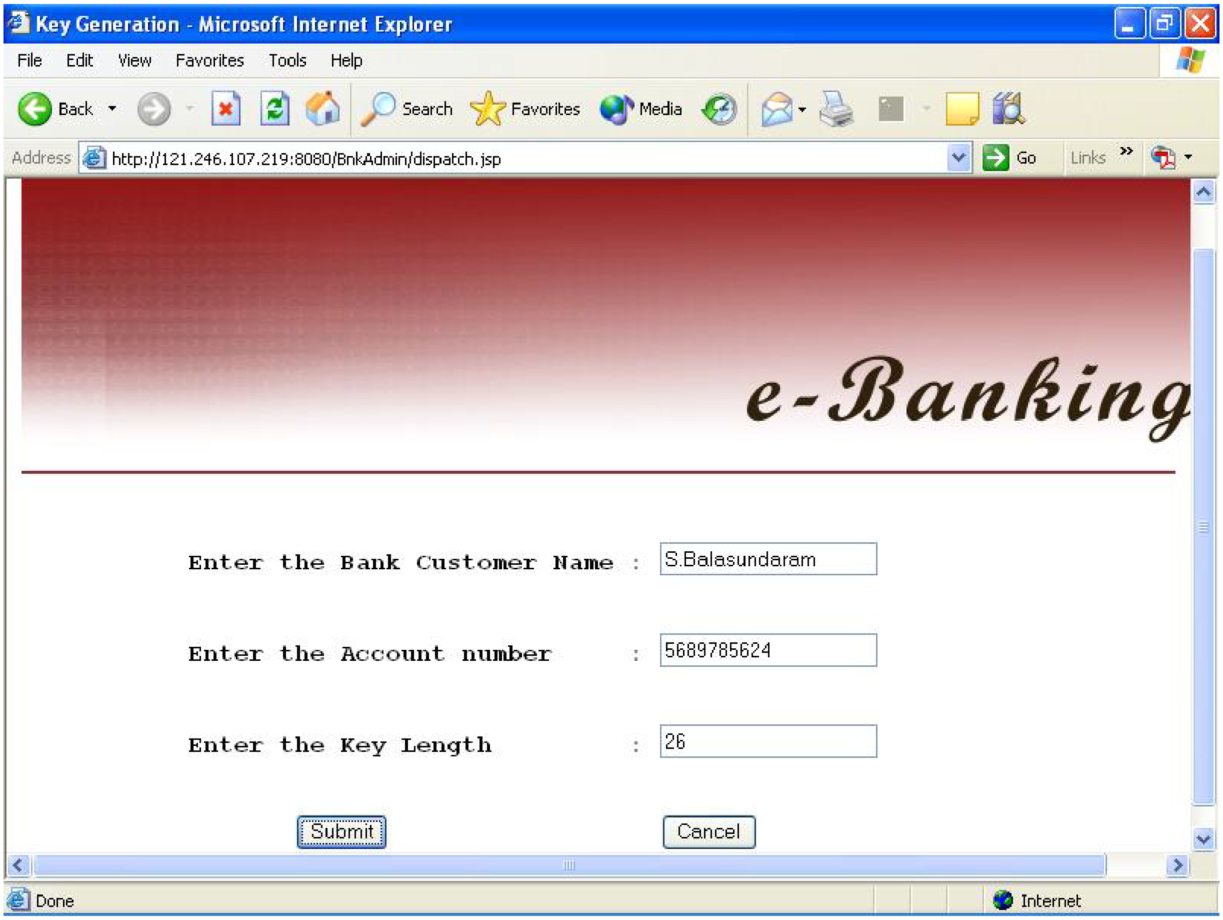
Description:

This screen provides the two main options to perform the following:

(i) To generate the Hyperelliptic curve of genus 2 and Divisor

(ii) To generate the Private and Public keys for the bank customers.

This page is accessed and executed by the bank administrator.


Key Generation - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address http://121.246.107.219:8080/BnkAdmin/dispatch.jsp
Go
Links
e-Banking
Submit
Cancel
Done
Internet

Enter the Bank Customer Name : S.Balasundaram

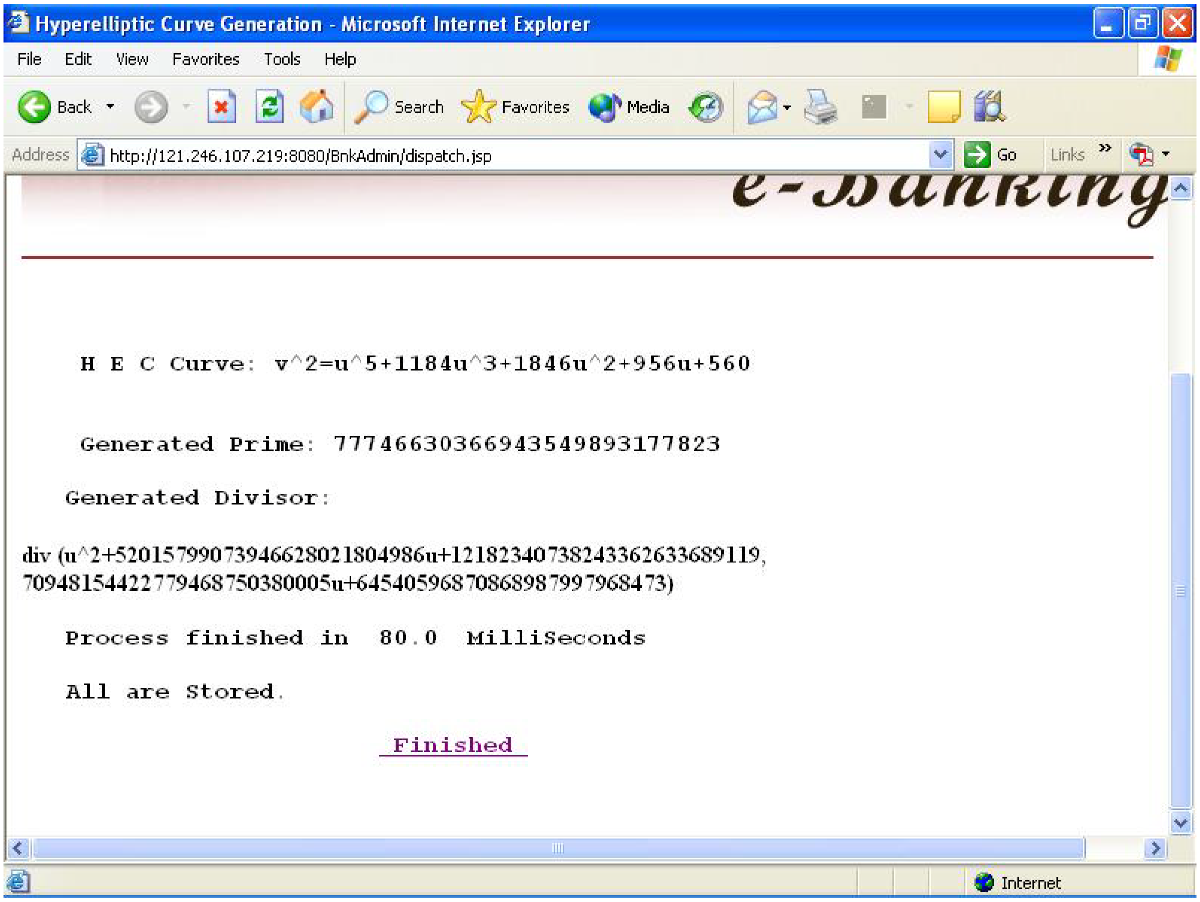
Enter the Account number : 5689785624

Enter the Key Length : 26

Figure 5: Keys Generation - Initial Screen

Description:

This screen is used to accept the primary input for generating the private and public keys for each bank customer. The *key length* input refers the length of the private key to generate. Here, 80-bit is considered for the private key length.



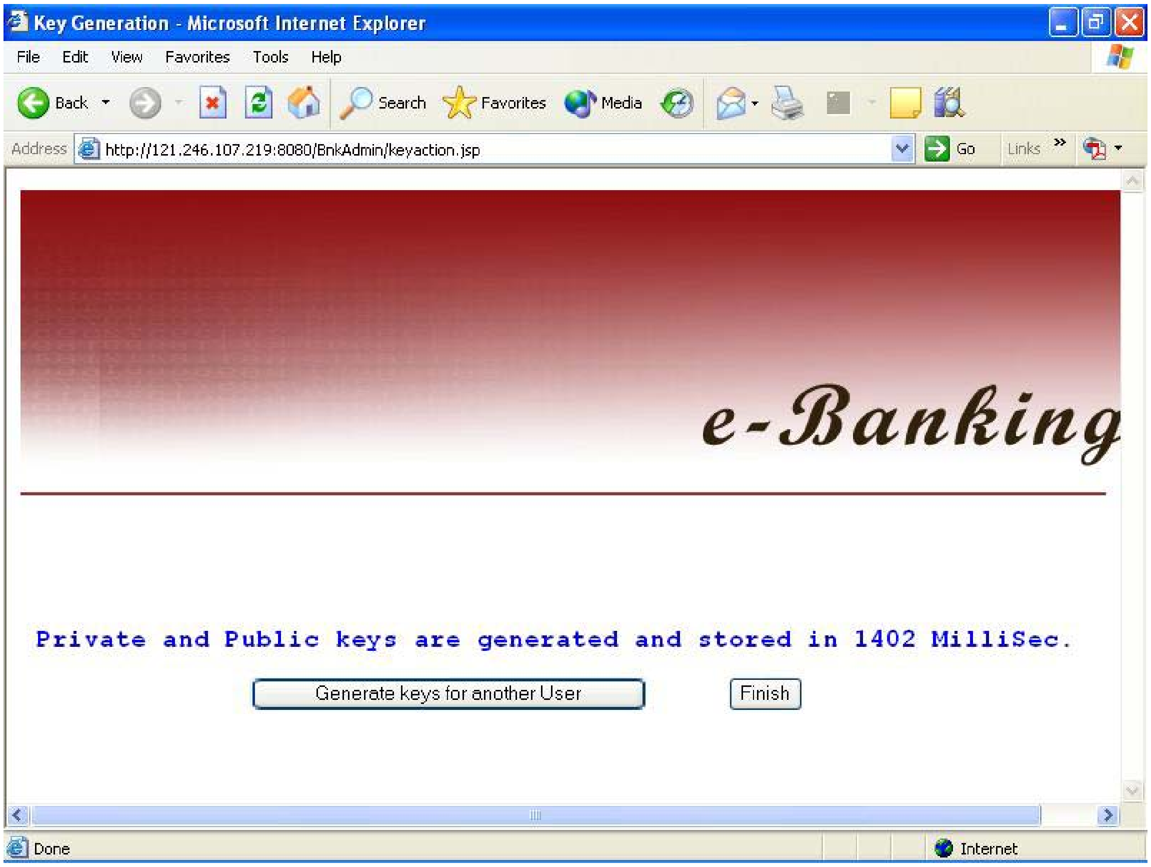
Hyperelliptic Curve Generation - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://121.246.107.219:8080/BnkAdmin/dispatch.jsp
Go Links
e-Banking
Internet

```
 H E C Curve: v^2=u^5+1184u^3+1846u^2+956u+560

 Generated Prime: 777466303669435498931 77823

Generated Divisor:

div (u^2+5201579907394662802 1804986u+1218234073824336263 3689119,
709481544227794687503 80005u+6454059687086898799 7968473)

Process finished in  80.0  MilliSeconds

All are Stored.

                    Finished
```

Figure 6: Generated HEC and Divisor Screen

Description:

The generated hyperelliptic curve of genus 2, prime number and the divisor are shown in the above screen. These are stored in the *Global_Parameters* table and are accessed during the key generation, encryption and decryption process.


Key Generation - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
Go Links
e-Banking
Generate keys for another User
Finish
Done
Internet

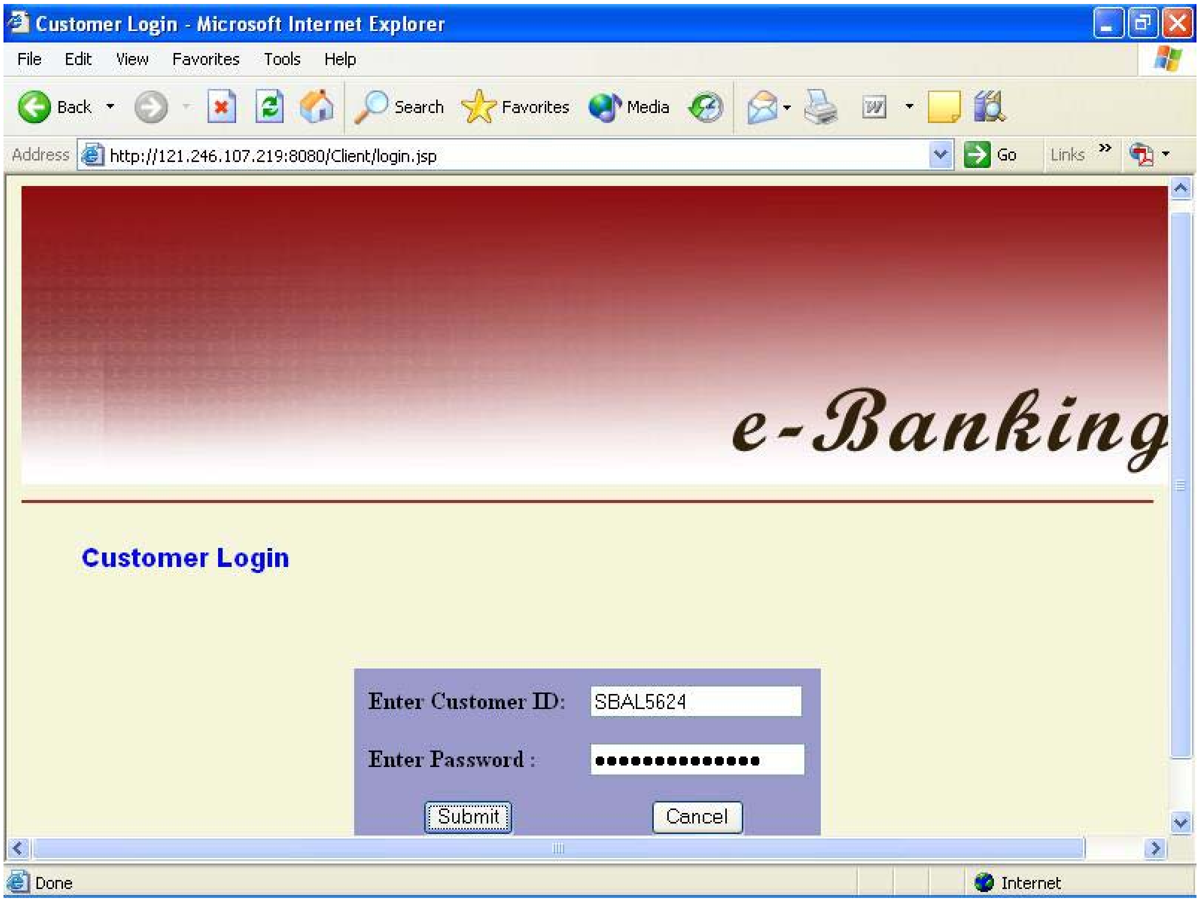
http://121.246.107.219:8080/BnkAdmin/keyaction.jsp

Private and Public keys are generated and stored in 1402 MilliSec.

Figure 7: Private and Public Keys Generation Screen

Description:

This screen intimates the status of the keys generated. The key generation method using HEC is executed to generate the keys. The generated keys for each bank customer are stored in the *User_keys* table. For each bank customer as well as the bank, there must be a unique pair of key. These keys are mainly used at the time of encryption and decryption.



http://121.246.107.219:8080/Client/login.jsp

Customer Login

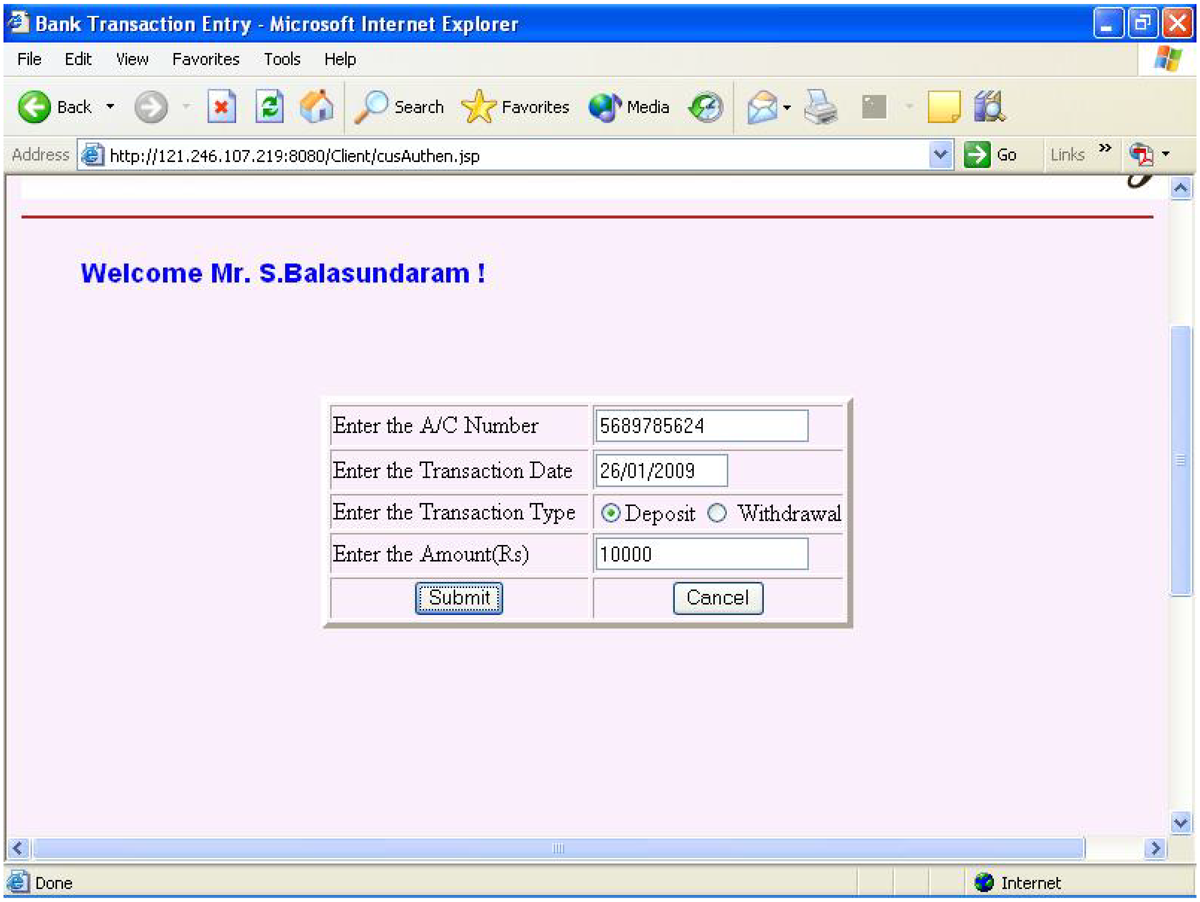
Enter Customer ID: SBAL5624

Enter Password : ●●●●●●●●●●●●●●●

Figure 8: Customer Login Screen

Description:

The above screen is used to authenticate the bank customer. Each customer has to enter customer ID and password. These data are encrypted using bank's pubic key and is sent to the application server for verification. Once the application server decrypts and obtains the original data, it is verified with the data which is already stored in the *Customer_Details* table. If both are identical, the application server allows the customer to view the *transaction entry screen,* otherwise displays an error message.


Bank Transaction Entry - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
Go
Links
Done
Internet

http://121.246.107.219:8080/Client/cusAuthen.jsp

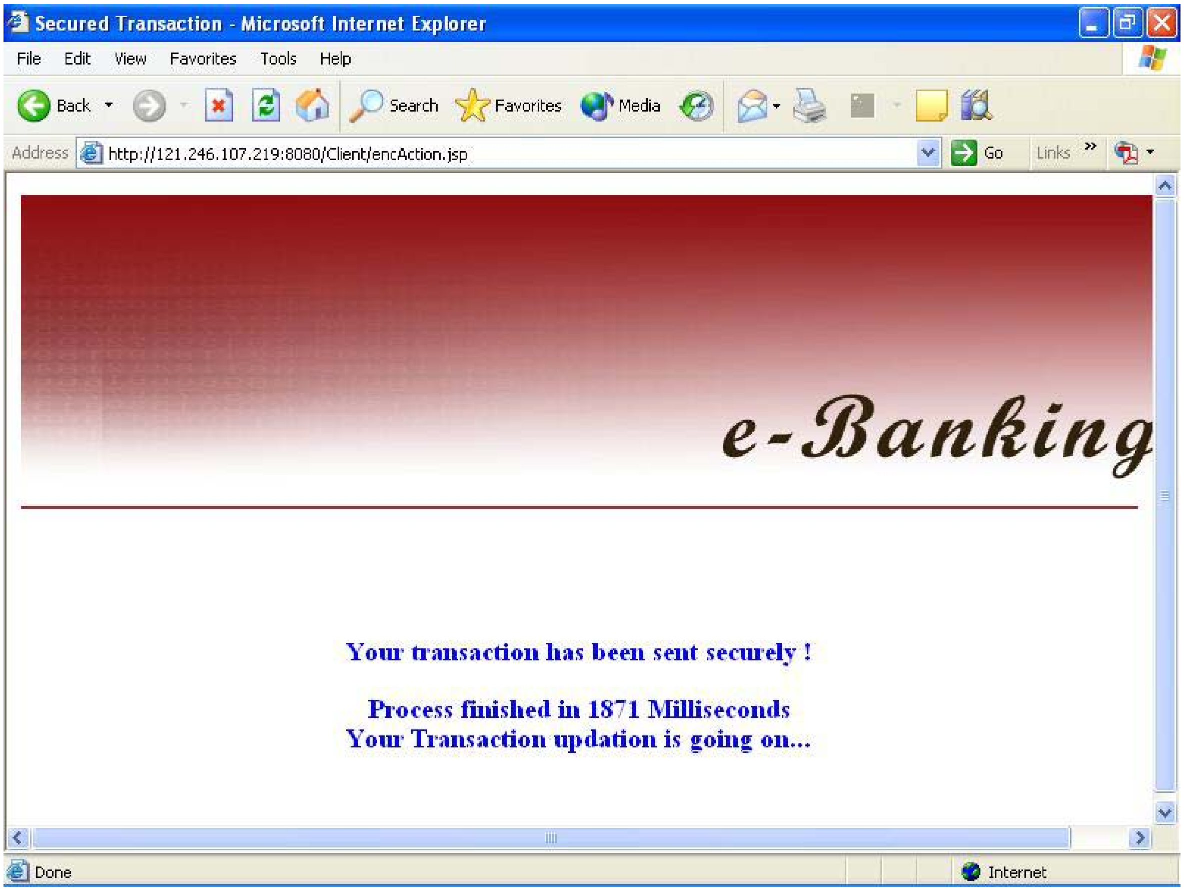
Welcome Mr. S.Balasundaram !

Enter the A/C Number	5689785624
Enter the Transaction Date	26/01/2009
Enter the Transaction Type	● Deposit ○ Withdrawal
Enter the Amount(Rs)	10000
Submit	Cancel

Figure 9: Bank Transaction Detail Entry Screen

Description:

Bank Transaction Details Entry screen is an important screen of this application. This screen allows the user to enter the sensitive transaction details. After the submit button is pressed, the encryption process starts.


Secured Transaction - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://121.246.107.219:8080/Client/encAction.jsp
Go Links
e-Banking
Done
Internet

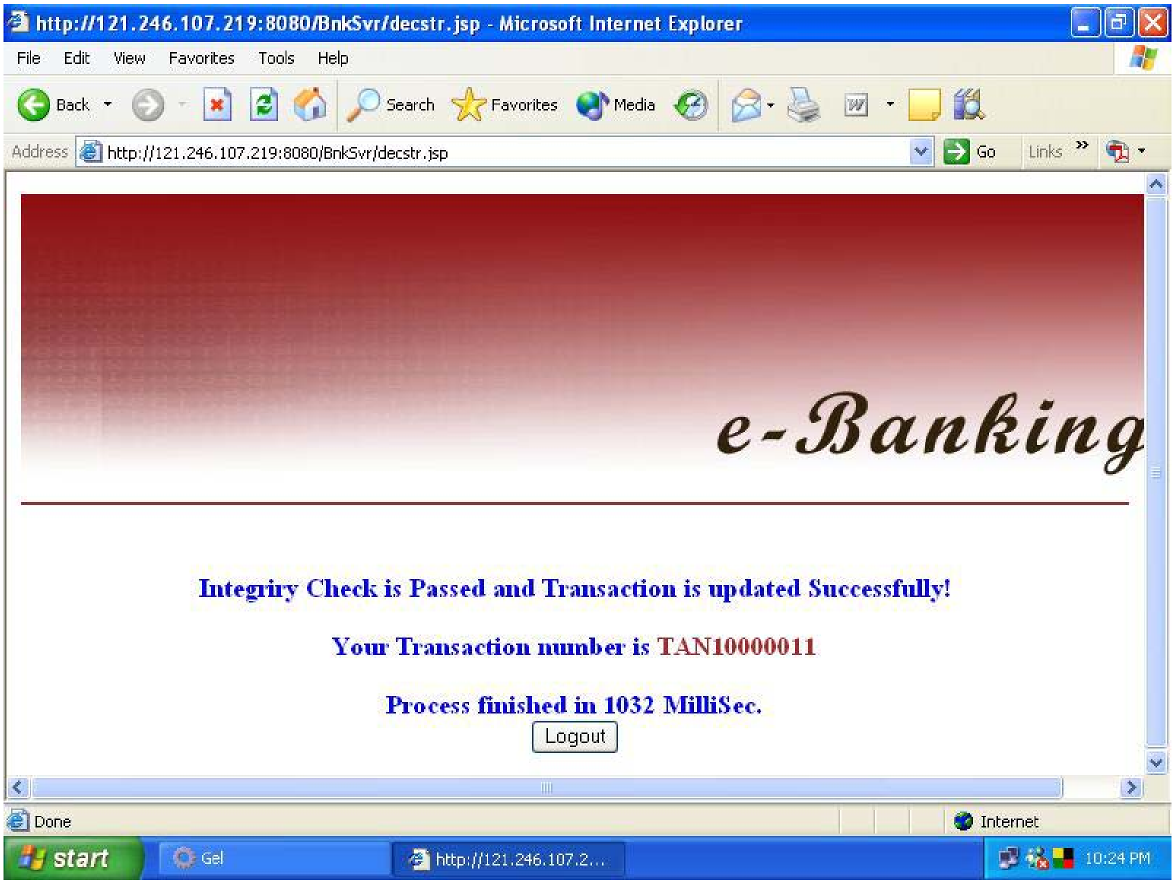
Your transaction has been sent securely !

Process finished in 1871 Milliseconds
Your Transaction updation is going on...

Figure 10: Transaction Encryption - Screen

Description:

The transaction data are encrypted using the bank's pubic key and is submitted to the application server in a secured manner. Here, the integrity of the data is maintained using the message digest and is encrypted using HECC. Both the encrypted transaction and encrypted message digest are transmitted to the application server for further process.



http://121.246.107.219:8080/BnkSvr/decstr.jsp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address http://121.246.107.219:8080/BnkSvr/decstr.jsp
Go
Links
e-Banking
Logout
Done
Internet
start
Gel
http://121.246.107.2...
10:24 PM

Integriry Check is Passed and Transaction is updated Successfully!

Your Transaction number is TAN10000011

Process finished in 1032 MilliSec.

Figure 11: Transaction Decryption - Screen

Description:

In the application server side, the received encrypted data is decrypted using HECC. The encrypted transaction is decrypted and it is subjected to MD5 hash algorithm for the integrity verification. Subsequently, the received encrypted message digest is decrypted. If both the computed message digest and the received message digest are identical, the transaction detail is stored in the *Trans_Details* table and the transaction number (TAN) is displayed. Otherwise, the error message “The integrity gets failed and the transaction is aborted” is displayed to the customer.

REFERENCES

Barclays Bank (2006), "*Business Internet Banking – Security and Confidentiality*", Item Ref: 9901713COM. May 2006.

Barker E, Barker W, Burr W, Polk W, Smid M, (2007), "*Recommendation for Key Management - Part 1: General (Revised)*", NIST Special Publication 800-57, March 2007.

Ganesan R, Gobi M, Dr. Vivekanandan K, (2008), "*Elliptic and Hyperelliptic Curve Cryptography Over Finite Field F_p", i-Manager's Journal on Software Engineering, Vol. 3, Issue No.2, October-December, 2008, pp 52-50, ISSN-0973-5151.

Ganesan R, Dr. Vivekanandan K, *"Performance Analysis of Hyper-Elliptic Curve Cryptosystems over Finite Field F_p for Genus 2 and 4",* International Journal of Computer Science and Network Security (IJCSNS) Vol.8, No.12, December 2008, pp 415 – 418.

Hiltgen A, Kramp T, Weigold T, (2006), "*Secure Internet Banking Authentication",* IEEE Security and Privacy, Vol. 4, No.2, 2006.

Dr. Janakiraman VS, Ganesan R, Gobi M, (2007), *"Hybrid Cryptographic Algorithm for Robust Network Security",* The International Congress for Global Science and Technology (ICGST), CNIR, Vol. 7, Issue 1, July 2007.

Liao Ziqi, Cheung M, (2002), *"Internet-based e-banking and consumer attitudes: An empirical study",*Information and Management, vol. 39, Issue 4, January 2002, pp. 283–295.

Liao Ziqi, Cheung M, (2003), *"Challenges to Internet E-Banking",* Communications of the ACM, Vo. 46, No. 12, December 2003, pp. 248-250.

Osama D, Phu Dung Le, Srinivasan B, (2007), "*Security Analysis for Internet Banking Models",* Eigth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNDP 2007, vol.3 , July 30 2007 – Aug 1 2007, pp 1141-1146.

Seitz J, Stickel E, (1998), "*Internet Banking – An Overview",* Journal of Internet Commerce and Banking, Vol.3, No.1, 1998. http://www.arraydev.com/commerce/JIBC/9801-8.htm.