E-COMMERCE PAYMENT AUTHENTICATION SYSTEM WITH SIM CARD

Rico Andropurnomo Tanaya[1)]

Abstract - Smart card is a device, which used for multi purpose needs to secure content and information. This device has many forms, for example like cards, which used to secure banking transaction, SIM card, which used to store operator subscriber's information as well as embedded smart card which used for many IOT (Internet Of Things) devices. The purpose of this research is to test whether SIM card can be used to secure credit card data that besides to store data, SIM card can also used to do verification and encryption. This research uses multiple standardizations as a base to build the prototype simulation that is PA-DSS and ISO 7816. PA-DSS standard used as a base to build client side and server side application and ISO 7816 is a protocol interface for smart card. With this research, I hope that we can see clear that it is possible to build a alternate payment system by using SIM card for storing purpose and credit card verification.

Key words: PA-DSS, ISO 7816, Smart Card, SIM Card, Internet Of Things

1. INTRODUCTION

Mobile devices growth in whole word rising exponentially as the technology become more advance. Mobile technology advance pushing new method of payment transaction that is e-commerce. Mobile commerce can be described as the ability of transaction by using mobile devices like cellphone, PDA, and computer. Mobile commerce use electronic transfer technology to do payment. Electronic transfer use mobile devices to pay transactions which issued by user.

2. LITERATURE REVIEW

First time smart card been introduced in 1968 by Helmut Gröttrup and Jürgen Dethloff, is a card which designed to store data securely. Smart card, chip card, or ICC (Integrated Circuit Card) is a card which has integrated circuit and only has ordinary size card. Smart card become a part of mobile technology and media which is used to sotre many informations efitiently. The beginning of smart card creation is because many security problem that occur in storing sensitive data. Java card which is used in this research is a SIM card look of java card. SIM card used by telecomunication company as a credential and place to store subscriber's identitiy. In e-commerce world, a secure transaction is a must. The point of online transaction is a user can order goods easily and as well pay securely. Now in online transaction when using credit card as payment, there are still some downsides.

These are some fraud with credit card:

1. phisically stollen credit card
2. steal credit card data by hacking merchant's websites or payment gateway providers.
3. Data stolen by web scam.

Problems that show up by using online transaction with credit card:

1. credit card information store in many places where this behaviour can be misused and data which the nature is secret can be stolen.
2. The need to enter credit card data everytime we want to do transaction.

3. RESEARCH METHOD

3.1 METHOD

This research is conducted because of the curiosity of writer to make a alternative of payment with credit card but with different

[1] Master of Information System Management, Binus Graduate Programs, Bina Nusantara University
email: rico.andro@gmail.com

verification method which is creating token and use SIM card.

SIM card choosen as a helper media because it is proven secure to store data, do verification, and token creation as a multi factor authentication system so there is no need to enter credit card number while doing payment.

Bank centric business model is choosen by writer as a base to create system architecture. This bank centric model choosen because bank and operator will act as service providers, data store providers, and do verification.

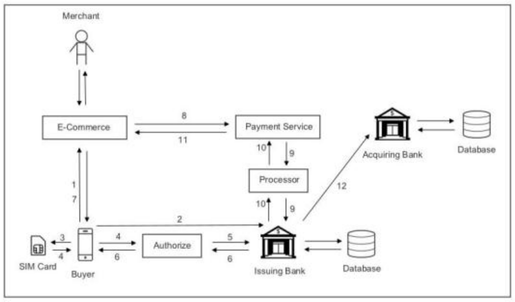

Fig 1: Proposed System Architecture

The steps:

1. Merchant and buyer do a transaction
2. Buyer request token to bank side
3. Through application, buyer do authentication in his mobile device by filling the password
4. After buyer authenticated, data encrypted and passed to bank side.
5. Bank decrypt the data and do verification
6. After buyer verified, bank send back token data to be used in payment
7. Buyer enter token number at merchant's website to pay
8. Merchant's website pass the data to payment service provider
9. Payment service provider will do verification to issuing bank
10. Issuing bank will do verification and respond back the status to payment service provider
11. If the status is ok, then payment service provider tell the e-commerce website that transaction is successful
12. Issuing bank will send the money that is been used by the user to merchant's bank.

The differences is with the archcitecture design. In the proposed design, first verification happen at user's smartphone by using the help from SIM card. Second verification is at server side by using token which is requested by user when SIM card verification has passed.

3.2 SIM CARD

These are the specifications from SIM card used for this research. This SIM card only support DES and 3DES encryption.

Hardware:

- CPU: 32-bit ARM
- Internal clock: 7.5MHz/ 15MHz/ 30MHz
- Memory: Flash 280KB
- RAM: 9KB
- Algorithm: DES/3DES
- ISO/IEC 7816 interface

Software:

- *Proprietary smart card OS* made only for this card.

3.3 CARD PERSONALIZATION

Every cards have different way to personalize. Based on the specification of the card that used for this research, picture below show the flow that is need to followed in order to do personalization to SIM card.

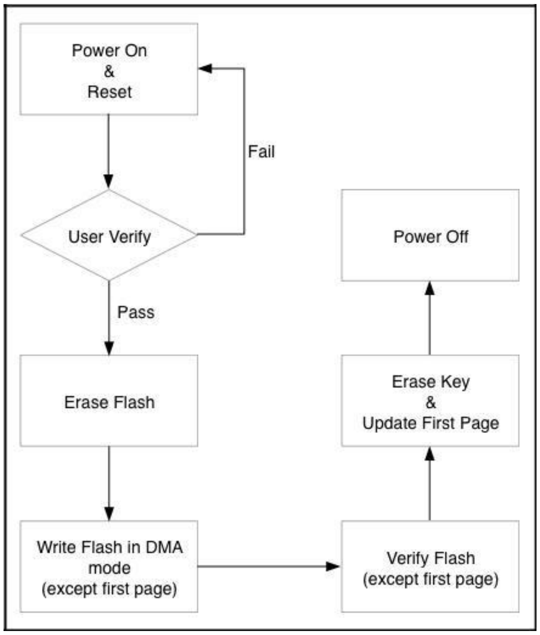

Fig 2: SIM Card Personalization Process

3.4 CLIENT DAN SERVER

Spesifikasi dari *hardware* dan *software* yang digunakan dalam implementasi *client* dan *server* adalah:

***Hardware*:**

- CPU: 2 core Intel i5 2.7GHz
- RAM: 8GB

- Disk space: 128GB

***Software*:**

- OS: Apple OSX 10
- Environment: Ruby On Rails
- Database: MySQL 5
- Web Server: Passenger

Programming:

This research use Ruby On Rails for programming language and APDU that follow ISO7816. Ruby On Rails programming laguage is used for server and client side where their purpose are to simulate verification and token generation as well as payment verification.

3.5 SYSTEM ARCHITECTURE SERVER

Server stored data needed for verification and authentication, like generated token data which later will be used for payment verification while shopping at e-commerce websites. All data stored in server are encrypted, this need to be done because only user know his own data, even database administrator cannot know what the data. Server also has obligation to send data first time to SIM card.

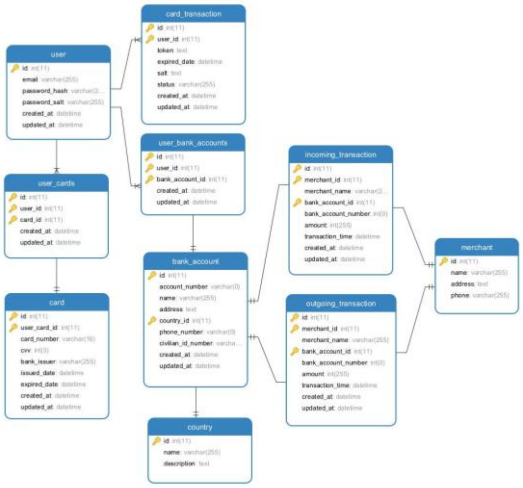

Fig 3: Server Application ERD

Client

Application in client side do not store data because like what written in PA-DSS standarization where data do not stored and like the purpose of this research that is user data stored at SIM card. The way that is used by client to establish a communication with server is by using REST technique, and to establish communication with SIM card with ISO 7816 standard, there are protocols which provided by mobile device in this matter is cellphone and the os that it use (iOS or Android).

USER AUTHENTICATION PROCESS

User authentication process is needed in order to use this application before menu for token request can be selected. User has to fill email and password, which later will be sent to server for login purpose and open session. This session also will be used to check the card which user has.

3.6 SIM CARD VERIFICATION PROCESS

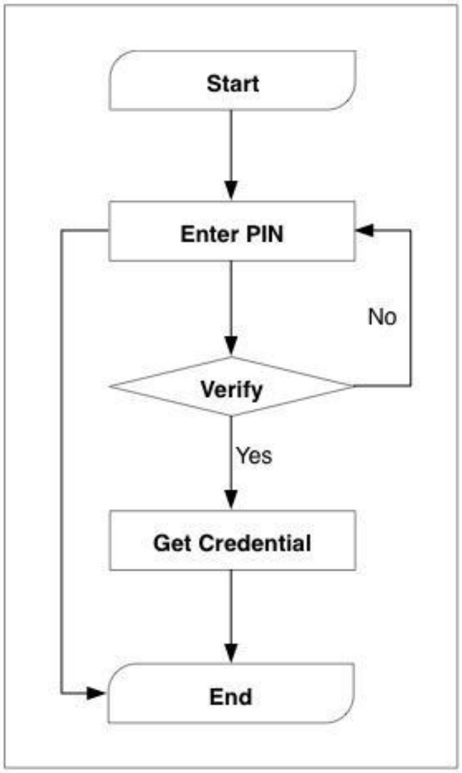

Fig 4: SIM Card Verification Process

Steps to do authentication are:

1. User will do PIN verification in his mobile device to use the service.
2. After verified, by using application user can grant access to credential where later this will be used as digital signature to get token.
3. The way to access UICC (Universal Integrated Circuit Card) can be done with Open Mobile API which is already installed in each mobile OS.

Users are required to enter a PIN to verify the SIM card to retrieve the data. PIN is inputted in the form of a pin credit card. In this research has been made an example PIN is 12345678 so that when the user enters a PIN, then the client application will convert to hex and forward the data to the SIM card with the command to verify CHV. The structure of command verify:

CLA	INS	P1	P2	Len	DATA	SW
A0	20	00	01	08	3132333435363738	90 00

This command is a command to verify CHV for data to be accessed is data protected and in accordance with the access condition that has been discussed in the section above, verify CHV1 need to read the data. Instructions verify (20) with a value of P2 (01) which is to verify CHV1, and requires 8 bytes of data (08).

3.7 DATA RETRIEVAL PROCESS ON SIM CARD

When the verify stage CHV has been successful, the next step is the retrieval of data from SIM card. Collecting data on the SIM card data retrieval is almost similar to the computer in general, is to open the directory where the data is stored, and then select the file containing the data. Below are the stages made from select directory and select the file containing the credit card data.

1. Select DF MyCard

CLA	INS	P1	P2	Len	DATA	SW
A0	A4	00	00	02	7F30	90 00

Before you can read the data in the file 7F40, it first must select the directory of the file. An instruction from select (A4) requires a data length of 2 bytes (02) a value EFID, and the data that is EFID which want to be selected (7F30).

2. Select EF

CLA	INS	P1	P2	Len	DATA	SW
A0	A4	00	00	02	7F40	90 00

Once the directory 7F30 been selected, then we can access the files contained therein is 7F40.

3. Read binary

CLA	INS	P1	P2	Len	DATA	SW
A0	B0	00	00	64	-	90 00

Once the file 7F40 which is the location of files which store data has been selected, the next step is to read the data. Instructions are B0 and expect output data is 64 bytes long.

3.8 DATA SENDING PROCESS

The data obtained from the SIM card when it has finished the data retrieval process contains data that has been encrypted with DES algorithm. Key to decrypt located in the server, and will be decrypted when the data is up to the server. This is done so that the data is secured so that if there are outsiders who want to steal the data, they cannot read it because the data obtained is encrypted.

The data obtained from the SIM card contains a credit card number and CVV number and an expiration date that has been encrypted. This data is then sent to the server to do the verification and manufacture token.

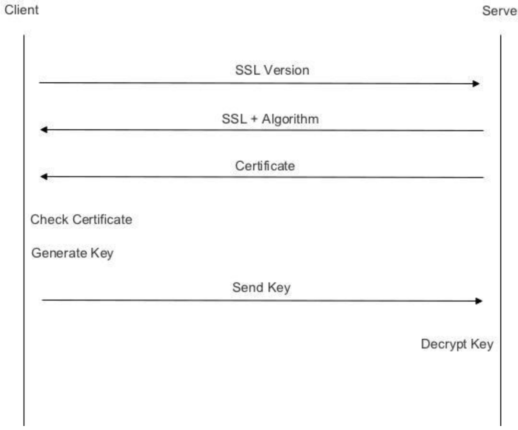

Fig 5: Handshake Process

Steps to make public key able to be sent securely is by using certificate.

1. Client application sent information containing SSL version used and symmetric algorithm used, then server will sent back the SSL version with algorithm used.
2. Server sent certificate. Client application check the certificate with its certificate in order to check whether the CA(Certificate Authorities) sent by a correct side.
3. If valid, application will make key for one session and encrypt it with server's public key got from certificate and send back the key to server
4. Server receive the data and do decryption. The key and the symmetric

algorithm will be used in the running session.

3.9 SERVER VERIFICATION PROCESS

The server do second verification before making a token that is by using the key to decrypt and check whether the received data is valid.

The process is carried out to validate the server are:

1. Data Decryption with DES algorithm
2. Check the card, whether the information provided is valid and belong to the appropriate user.

Server will do verification process based on incoming data, whether user ask for it is the valid user as the data owner.

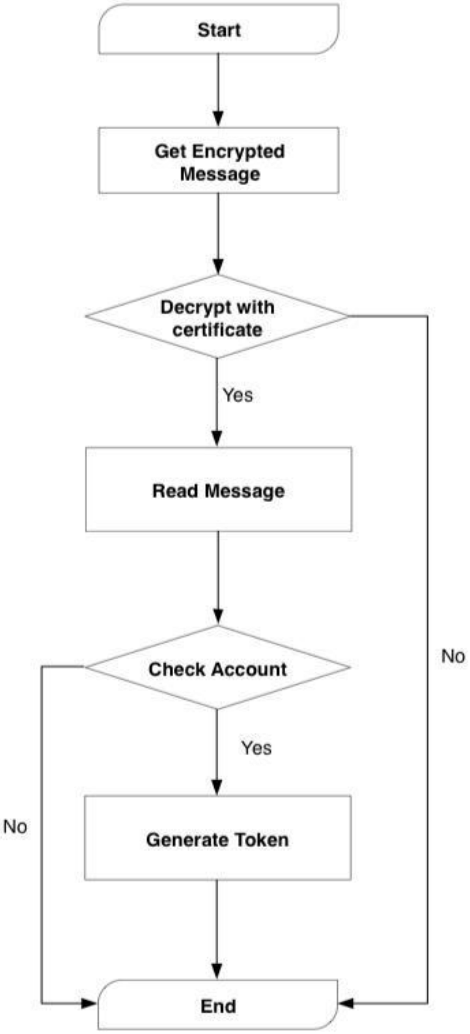

Fig 6: Verification Process

Steps to do verification process are:

1. By using certificate which is given to client, then server can do decryption by using private key that belongs to server and client's public key.
2. After decrypted and the message can be read, server will do checking to user's account.

If user verified, temporary token for user created and stored into database. This data will be used as payment verification

3.10 TOKEN GENERATION PROCESS

After the server do second verification by using the data received, token generation requires several variables:

1. Token

   Creating a random hex value over the 10 bytes by using Secure Random Generator. This is a token prior to encryption.

   A token with 2904628757317be1cdc4 value has been created. This value will be stored in a database, which will then be used for the verification process by the payment service provider.
2. Salt

   Salt is used as a random data used for additional input in the manufacture of a password, in this case the token. A salt with a length of 64 bytes:
   30930864f57285e5fb3a64128808ac3dda2230b2722a434910305a5315ca9dacc5de291dc426cd8d9fd32f2072657c1530b869f8dc707785b699e9f8cd854c02
3. *Key*

   Key made with key generator using PIN and salt.
   "\xD9}5Qh!\xD8\xD7\xA3W\xED=\xC9\x844\xEE8\x17\xF5\xD5w*\xEB\b\x12W\x87\xE0\xEA\x97\xBD\xCB\xEB\x85\x95e\xBD\xEF7\xF1\xB3\xFB\x8A5-\xDA\xC9gl*'k\v\xB5\x8F^\x02\xE5tY\x01@\x8B\x98"
4. *Message encryptor*

   Message encryptor used to encrypt cipher text by base64 encoding which is required when you want to send or save the data to a place that we unsured about it. This method is used so that when the data is taken by the party were not proper, the person cannot see the message sent.

   The result of the creation of message encryptor using a key that was created in the previous process:
   #<ActiveSupport::MessageEncryptor:0x007fca5269e028

```
 @cipher="aes-256-cbc",
@secret=
"\xD9}5Qh!\xD8\xD7\xA3W\xED=\xC9\x
844\xEE8\x17\xF5\xD5w*\xEB\b\x12W\x
87\xE0\xEA\x97\xBD\xCB\xEB\x85\x95e\
xBD\xEF7\xF1\xB3\xFB\x8A5-
\xDA\xC9gl*'k\v\xB5\x8F^\x02\xE5tY\x01
@\x8B\x98",
@serializer=Marshal,
@sign_secret=nil,
@verifier=
#<ActiveSupport::MessageVerifier:0x007
fca5269dec0
@digest="SHA1",
@secret=
"\xD9}5Qh!\xD8\xD7\xA3W\xED=\xC9\x
844\xEE8\x17\xF5\xD5w*\xEB\b\x12W\x
87\xE0\xEA\x97\xBD\xCB\xEB\x85\x95e\
xBD\xEF7\xF1\xB3\xFB\x8A5-
\xDA\xC9gl*'k\v\xB5\x8F^\x02\xE5tY\x01
@\x8B\x98",
@serializer=ActiveSupport::MessageEnc
ryptor::NullSerializer>>
```

5. *Encrypted data* / token
   Message encrypted and given a mark / sign in order to avoid padding attacks or attacks carried out by adding a message / bytes in the message with the cypher.

   ```
   alpkVmdmYitUVlQ1cVNKaHBMMHBwO
   ElHaFVTR25SRHF1a1MzbzI1ZGszWT0t
   LWVwM3Z4V3BPNUtmMXM1QWZMV3l
   YblE9PQ==--
   332446ec76079e31614a3f6c707525cc2fe
   dfc74
   ```

   Values above are a token value that has been encrypted and sent back to the client. Only a client with a valid key can read the original token value.
   Tokens which are made have a maximum expiration time of 1 day from the time its created. The results will be stored in a database, which will then be used by payment service providers when users make a payment using the token.

4. RESULT AND DISCUSSION

Table 1: Architecture Comparisson

Architecture	**JASA**	**SET & iKP**	**LSM**	**SA2pMP**	**Our**
Field					
Targeted to M-Payment	v	no	v	v	v
Targeted to 2-Party	v	(1) KP	v	v	v
Cryptography Algorithm	AES	RSA	RSA	ECDSA, AES	DES, AES, RSA
Authenticaiton Strategy	SFA	SFA	SFA	MFA	MFA
Non_Repudiation	No	v	v	v	
Java ME Enabled	v	-	-	v	-
Computational Requirements	Light weight	Light weight	Light weight	Light weight	light weight
The 3rd-Party Involvement	low	high	high	Medium	high
Using secure element	no	yes	no	no	yes

Advantage of designed system

1. Memory smart cards are non-volatile, programmable read-only memory.
2. Card can check the password entered by the user with the password stored in the memory card so it does not need to reveal the password to the system or outside the system using smart cards outside to check, such as mobile devices.
3. Can be checked without having to be online.
4. Smart cards more difficult to be vandalized and stolen data because the CPU and the memory card cannot be accessed except by the owner and creator of the card.
5. The mobile device does not store anything, just act as a channel data.
6. No need to carry a credit card, only need to remember a pin.

System limitation

Just like many other industries are difficult to change something that is already running or creating new alternatives, in this case the payment. Difficulty to introduce and make the new system to be accepted by society. As an example of the payment by smart card and pin is the safest way, but the first bank was afraid that if the customer can not remember the pin so that they do not use a pin to make a transaction, but sooner or later, use pin becomes mandatory in every transaction.

Result

1. DES encryption algorithm is limited because of the limitations of the SIM card used for testing only supports DES and 3DES algorithm.
2. The verification and encryption of data that do SIM card to be sent to the server requires a fast time and almost instantly ie below 1 second.

5. CONCLUSIONS

1. Using the SIM card as the storage media and credit card verification proved to be safe because it uses several symmetric and asymmetric encryption methods such as DES and AES.
2. Nature of the smart card memory is non-volatile and availability of the CPU so that the process of validation PIN does not require the process of mobile devices.
3. SIM card proven can be used as a tool to store credit card data, and authentication.
4. The ability of the system to perform a series of encryption and decryption with DES and AES method, and performance in making the token is very good because the time required for the whole process less than 1 second.
5. Ability card encryption depends on the specifications of the SIM card. SIM card can only perform encryption based algorithms are supported, such as those used for this study only supports DES algorithm.
6. This study is limited only supports the storage of credit card information in the SIM card so it is not possible for the user to store the entire credit card number is held when there are more than one.
7. If the credit card data is blocked because of a mistake entering a PIN more than three times, then the user must ask for help from the bank to do the unblocking and change the PIN.

6. REFERENCES

[1] C. Considerations, “Secure Authentication for Mobile Internet Services Table of Contents,” no. December, pp. 1–23, 2011.

[2] A. Data, “Shared key TLS usage within Ua interface,” no. May, pp. 1–7, 2004.

[3] Gemplus, Oberthur, and Schlumberger, “Over-The-Air (OTA) technology,” no. October, p. 6, 2003.

[4] V. Guyot, “Smart card, the invisible bullet,” *9th Eur. Conf. Inf. Warf. Secur. 2010, ECIW 2010*, pp. 80–87, 2010.

[5] International Organization for Standardization (ISO), “ISO7816-4 Organization, security and commands for interchange,” vol. 2005, 2005.

[6] A. Khalique, K. Singh, and S. Sood, “A Password-Authenticated Key Agreement Scheme Based on ECC Using Smart Cards,” *Int. J. Comput. Appl.*, vol. 2, no. 3, pp. 26–30, 2010.

[7] I. E. Liao, C. C. Lee, and M. S. Hwang, “A password authentication scheme over insecure networks,” *J. Comput. Syst. Sci.*, vol. 72, no. 4, pp. 727–740, 2006.

[8] S. I. Module and M. Equipment, “GSM Digital cellular telecommunications system Specification of the Subscriber Identity Module,” 1996.

[9] A. Pourali and M. H. Yektaie, “A Secure SMS Model in E-Commerce Payment using Combined AES and ECC Encryption Algorithms,” pp. 431–443, 2014.

[10] S. A. Procedures, “Payment Card Industry (PCI) Payment Application Data Security Standard Requirements and Security Assessment Procedures,” *October*, no. October, p. 73, 2010.

[11] Q. Requirements, “International Standard Iso / Iec,” vol. 25021, 2012.

[12] I. Sommerville, *Software Engineering*. 2010.

[13] C. Specification, “GlobalPlatform,” no. March, 2006.

[14] I. Standard, “INTERNATIONAL STANDARD ISO / IEC AMENDMENT 2 : Conservation of prefixes,” vol. 2008, 2008.

[15] G. S.Thompson, “PAYMENT SECURITY AND THE EMV CHIP TRANSITION by Gary Scott Thompson A Capstone Project Submitted to the Faculty of Utica College December 2015 in Partial Fulfillmentof the

Requirements for the Degree of Master of Science in Cybersecurity i," no. December, 2015.

[16] L. T. Thorsen, "Multi-factor Authentication using Secure Elements," 2016.

[17] C. Wohlin, D. Šmite, and N. B. Moe, "A general theory of software engineering: Balancing human, social and organizational capitals," *J. Syst. Softw.*, vol. 109, pp. 229–242, 2015.

[18] [18] S. Zanero, "Smart Card Content Security," *Dip. di Elettron. e Inf.*, 2002.