Fraudulent Internet Banking Payments Prevention using Dynamic Key

Osama Dandash
Faculty of IT,
Monash University, Melbourne, Australia
odan1@student.monash.edu.au

Yiling Wang
Faculty of IT,
Monash University, Melbourne, Australia
ylwan2@student.monash.edu

Phu Dung Leand Bala Srinivasan
Faculty of IT,
Monash University, Australia
Phu.Dung.Le@infotech.monash.edu.au, Bala.Srinivasan@infotech.monash.edu.au

Abstract **As the Internet becoming popular, many sectors such as banking and other financial institutions are adopting e-services and improving their Internet services. However, the e-service requirements are also opening up new opportunity to commit financial fraud. Internet banking fraud is one of the most serious electronic crimes (e-crimes) and mostly committed by unauthorised users. This paper presents a new dynamic key generation scheme that facilitates a fraud prevention mechanism. In the proposed scheme, a combination of a biometric feature such as a fingerprint [10] and smart card [6][11] is used to effectively confirm the users' identity and prevents illegal attempts. It also eliminates the need for storing a long-term shared key which makes the system insecure during transactions. We show that the new scheme is secure against various kinds of attacks.**

Keywords: **Internet Banking payment, key generation, fraud, payment systems**

I. INTRODUCTION

Internet banking use reduces costs and allows banking institutions to reach out to more customers. However, the e-service requirements are also opening up new opportunities to commit financial fraud[9]. A survey was done by UK Information Security Breach and the UK National Hi-Tech Crime Unit's recent report has evidenced this serious situation. It highlights that "online financial fraud is one of the most serious e-crimes and takes the lion's share of over 60% of e-crime costs"[5].

Billions of pounds are lost each year in the banking sector due to unauthorised users committing fraud through the exploitation of system vulnerabilities [5]. In addition, the opportunity to commit fraud online is higher than the manual ways of performing transactions as the transactions performed within a given time period. Failure to prevent unauthorised and illegal use can lead to financial loses and damage the reputation of financial institutions.

Various kinds of secure payment systems over the Internet have been implemented such as Secure Sockets Layer (SSL) [1][25] and Secure Electronic Transaction protocol (SET) [14]. However, several security issues related to exposing credit-card information have been reported. In SSL-based credit-card payment system [1], the credit-card information might be revealed, which leads to illegal and fraudulent actions. In SET [14] protocol [6], encrypted credit-card information is decrypted by the payment gateway and then forwarded to the issuer. A security problem may arise if the payment gateway and the issuer are different parties.

Moreover, the credit-card number is printed on the card which makes it visible to everyone and obtaining the client's information such as date of birth is not a difficult task. In addition, the credit-card number is considered reusable long-term and semi-secret information. It can be replaced by a secret that is only known to the client and the issuer as in Kungpisdan-Srinivasan-Le (KSL) protocol [29]. However, it still has to be transferred in every transaction. Thus, it is vulnerable to various kinds of attacks.

As a result, efficient security measures are needed to prevent fraudulent financial transactions performed by unauthorised users and to ensure transaction integrity. To achieve so, this paper introduces a new Dynamic Key Generation (DKG) scheme that overcomes the shortcomings of the existing systems and facilitates fraudulent Internet banking payments prevention. The new scheme uses advanced authentication technologies to identify users and prevent fraudulent attempts.

It also eliminates the need for storing long-term shared key which makes the system insecure against key compromise during transactions. In the proposal, the generation of each set of keys is based on randomly chosen preference keys. The higher the number of transactions performed the less chance the system has of being compromised. Our proposal also makes use of the advanced authentication technologies such as smart cards [6][11] and biometrics [10]. The shared keys generated from our proposal can be used either as authentication tokens or as the keys for encryptions and MAC (Message Authentication Code) operations.

This paper is organized as follows. Section 2 outlines existing key generation techniques and discusses their security issues. Section 3 introduces the proposed dynamic key generation technique. In section 4, we apply the proposed technique to enhance security of internet payment systems. Section 5 discusses and analyses the security of the proposed scheme. Section 6 concludes our work.

II. RELATED WORK

In this section, we outline two existing limited-use key generation techniques

A. Figures and Tables

Rubin et al. proposed an offline Disposable Credit Card numbers (DCN) generation technique [27]. The proposed method eliminates the need for traditional reusable long-term credit-card numbers. A DCN which is called a token as in [27] is generated from encrypting a set of payment-related information (called restrictions). The information includes the amount of payment, the merchant's identity, billing address in addition to a long-term shared key between the client and the issuer. For instance, a token T will be generated if Alice buys a 30-dollar book from Bob's store as follows:

$$T = \{\text{thirty-dollars-book-Bob's-store}\}_K$$

Where K is the long-term key shared between Alice and the issuer.

On receiving T, the issuer decrypts it using K. The proposed technique also deploys timestamp for replay and collision protection. If different payment information is encrypted by either the same key or by different keys then the collision may occur.

Although, Rubin et al. [27] argued about the system's security against various kinds of guessing attacks however, to some degree, encrypting using long-term shared key is vulnerable if an attacker obtained enough information and attempts to decrypt DCN. Compromising the long-term key will lead to system failure. Clients whose credit card information falls into the wrong hands won't wait until the fraud is being detected. Moreover, Rubin et al. [27] argued that more users and restriction will cause the encryption to become computationally expensive.

B. Kungpisdan et al.'s Approach

Kungpisdan-Srinivasan-Le (KSL) Protocol [29] was designed and implemented for security enhancement and protection of Wireless Internet payment using Credit Cards. The client connects to the merchant through the access point to perform an m-commerce transaction. In KSL architecture [29], the client sends the Value Subtraction Request to the payment gateway through the merchant instead of sending it directly to the merchant to minimize the number of connections needed. It is divided into two phases, where the client has to register with the merchant and send the merchant the master key in the first phase. And in the next phase the purchase takes place by generating a session key from the master key, which has been distributed between the client and the issuer when the client first registered with the issuer.

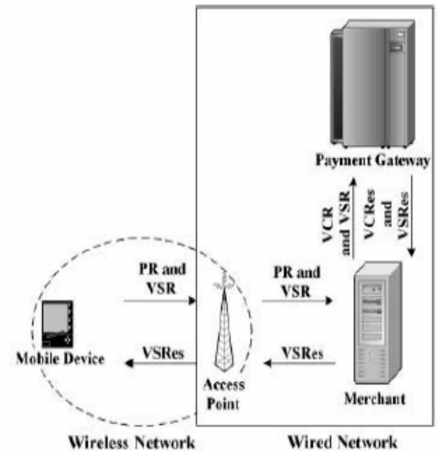

Figure 1. KSL Architecture

KSL [29] deploys a technique where a set of session keys can be generated from a master key using hashing and cyclic-shifting techniques. The generated keys will then be used in transactions. The set of Xi, where i = 1,…., n, shared between the client and the merchant can be generated as follows:

$$X_1 = h(\text{1-bit-shift-of-X}),\ X_n = h(\text{n-bit-shift-of-X}).$$

The set of Yi, where i = 1,…., n, shared between the client and the issuer can be generated as follows:

$$Y_1 = h(\text{1-bit-shift-of-(Credit card information, Y)}),$$
$$Y_n = h(\text{n-bit-shift-of-(Credit card information, Y)}).$$

The disadvantages of KSL key generation technique are as follows:

- It relies on credit card information, which can easily be stolen and used illegally
- It does not have the ability to identify the card holder
- It is unable to prevent attackers from performing fraudulent transaction

III. The Proposed dynamic key generation Technique

A. Notations

The following notations will be used in this scheme:

- C, B and BS: represent the client, bank and the bank server respectively
- $\{ID_C, ID_{BS}\}$: the set of identities of C (Bio_ID) and BS respectively
- ShK: represents Shared Keys
- ShS: represents Shared Secret, (e.g. C's date of birth) for identification verification
- SK: represents Session Keys
- TPass: transaction password
- (K*i*)s: represents Secondary Keys
- (r) represents Random number
- V: a calculated set of values (V_1, V_2, V_3) used in generating SK's
- VT: represents Value of Transaction, it equals hashed V ($h\{V_1, V_2, V_3\}$)
- N_C, N_B denotes statements
- X: the message
- *h*(x) represents the hash function
- {T, T_{ID}: the transaction and its identity including time and date
- TI: Transaction information
- TD: transaction descriptions, debit, credit or payment
- Amount: the amount and currency
- PI: payment information, which contains the smart card's information
- I: client's information such as address or contact details
- Yes/No: the status of authentication and transaction approved/rejected
- T_{ID}Req: the request for T_{ID}
- C, B $\models$ |X|: C and B believe X
- C $\overset{K}{\longleftrightarrow}$ B: C and B use a shared key
- C $\overset{X}{\Leftrightarrow}$ B: X is secret only known to C and B
- $\xrightarrow{K}$ C: C has a secret key.
- B $\triangleleft$ X: B sees X
- C |~X: C has sent the message X.

B. The Proposed Technique

The security of symmetric-key based systems relies heavily on the privacy of long-term shared keys. If the keys are revealed to an attacker, the security of the entire system will be compromised. This section presents a limited-use key generation technique in which the key used in each session does not rely on any long-term key so that the compromise of the long-term key does not affect the security of the system. Generating the session keys from a master key shared between parties is a possible solution to a compromised long-term key.

However, the key generation technique must be secure in that it must be difficult for an attacker to compute the master key from capturing the session keys. Figure 3.1 depicts our dynamic key generation.

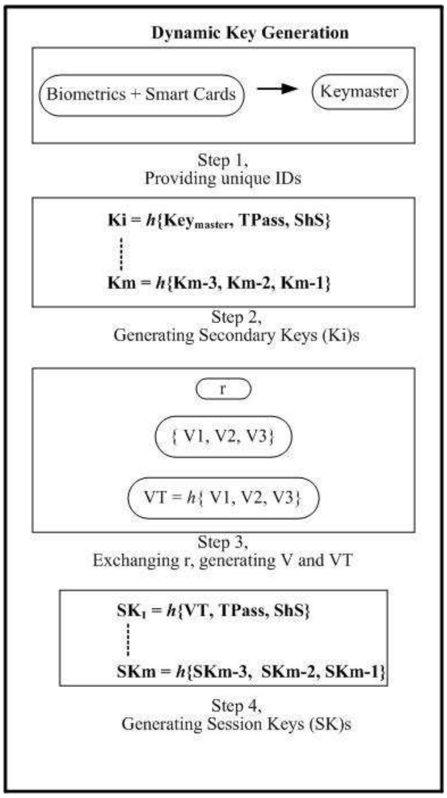

Figure 2. Illustrates DKG Scheme

In the proposed scheme, a combination of two or more advanced authentication methods such as fingerprint [10] and smart card [6][11] are used to effectively confirm the users' identity, enhance the transactions security and to protect payment details. The primary advantages of biometric authentication methods over other authentication methods are that they use real human physiological or behavioural characteristics to authenticate users. Such characteristics are generally permanent and it is not easy to change fingerprint, iris or other biometric characteristics.

B. Dynamic Key Generation

In the first step, a Key_{master} will be dynamically generated based on the user's unique identification to identify users and prevent attackers from performing fraud payments.

The second step shows the generation of Secondary Keys (Ki)s. (Ki)s are necessary to make the generation of Session Keys (SK)s more complicated and very hard to guess. They are an enhancement security step as they will be used to generate V values which will be one of the major factors in generating (SK)s.

The generation of (Ki)s relies on the combination of three factors; Key_{master}, Transaction Password (TPass) and Shared Secret (ShS).

TPass is a secret password. ShS is used as a security enhancement after the transaction password is confirmed and for identification. It can be of any value (e.g. C's date of birth). The generation of (ki)s occurs as follows:

$$K_i = h\{Key_{master}, TPass, ShS\} \quad (B1)$$
$$K_{i+1} = h\{TPass, ShS, K_i\}$$
$$K_{i+2} = h\{ShS, K_i, K_{i+1}\}$$
$$K_{i+3} = h\{K_i, K_{i+1}, K_{i+2}\}$$
$$\vdots$$
$$K_m = h\{K_{m-3}, K_{m-2}, K_{m-1}\} \quad (B5)$$

The generation of (K*i*) **(B1)** relies on the existence of the three factors, whereas the next generated keys eliminate one of them after each generation step. When (K_{i+3}) is generated, all the factors would have been totally eliminated and therefore it will be the first key used in transactions. Such an elimination mechanism provides a high security improvement and makes the generation of (K*i*)s more complicated for attackers to compromise.

The same shifting technique is applied for (SK)s generation as well. Also it can be seen that the elimination of each of the factors in (K*i*) **(B1)** illustrates the one-way hash property of the function in which the values of (K*i*)s don't depend on the values of subsequent keys. This makes it mathematically infeasible to calculate any of generated (K*i*)s value from first principle.

In step three, a random (r) (details will be presented in section 4.1) is chosen and used for the generation of V values:

$$\{Random(r)\}_i \quad (B6)$$

The indices (i)s can be used for identification confirmation and to the final number of key generation steps assigned to the transaction at the beginning. They also assure that the generation of (K*i*)s experience no complications. After the identity is confirmed, the generation of V values (V_1, V_2, V_3) will take place as follows:

$$V_1 = r \bmod (m-3)$$
$$V_2 = r \bmod (m-2)$$
$$V_3 = r \bmod (m-1) \quad (B7)$$

Where m-3, m-2 and m-1 are hashed values of the last calculated secondary key (Ki).

$$(m-3, m-2, m-1) = h(K_i, K_{i+1}, K_{i+2})$$

The generated V values will then be hashed to generate a Value of Transaction VT value, one of the pillars in generating (SK)s as follows:

$$VT = h\{ V_1, V_2, V_3\} \ (B8)$$

The generation of (SK)s relies on the combination of three factors; VT, TPass and ShS as shown in step four:

$$SK_1 = h\{VT, TPass, ShS\} \quad (B9)$$
$$SK_2 = h\{TPass, ShS, SK_1\}$$
$$SK_3 = h\{ShS, SK_1, SK_2\}$$
$$SK_4 = h\{SK_1, SK_2, SK_3\}$$
$$\vdots$$
$$SK_m = h\{SK_{m-3}, SK_{m-2}, SK_{m-1}\} \quad (B13)$$

This technique does not rely completely on any long-term shared keys. The greater number of used generated (SK)s will minimize chances to compromise the system. Moreover, the number of generated (SK)s must not exceed the number of required steps for any particular transaction. The limited-use shared keys can also be used as single-use authentication tokens as well as keys for encryptions or hash functions.

IV. THE PROPOSED INTERNET BANKING MODEL

Based on the proposed DKG scheme, an Internet banking payment fraud prevention model is been designed in addition to session key generation protocol as illustrated in figures 4.1 and 4.2. The main concept of this model is to:

- Strengthen the security of users' authentication,
- Protect the sensitive data during transactions
- And prevent any fraudulent attempts by generating random keys.

Next, more details regarding the proposed model and the protocol using the model is outlined.

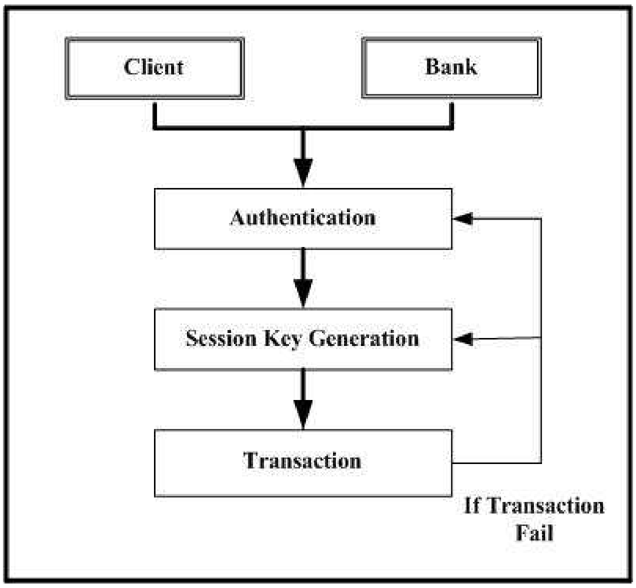

Figure 3. Illustrates the proposed model

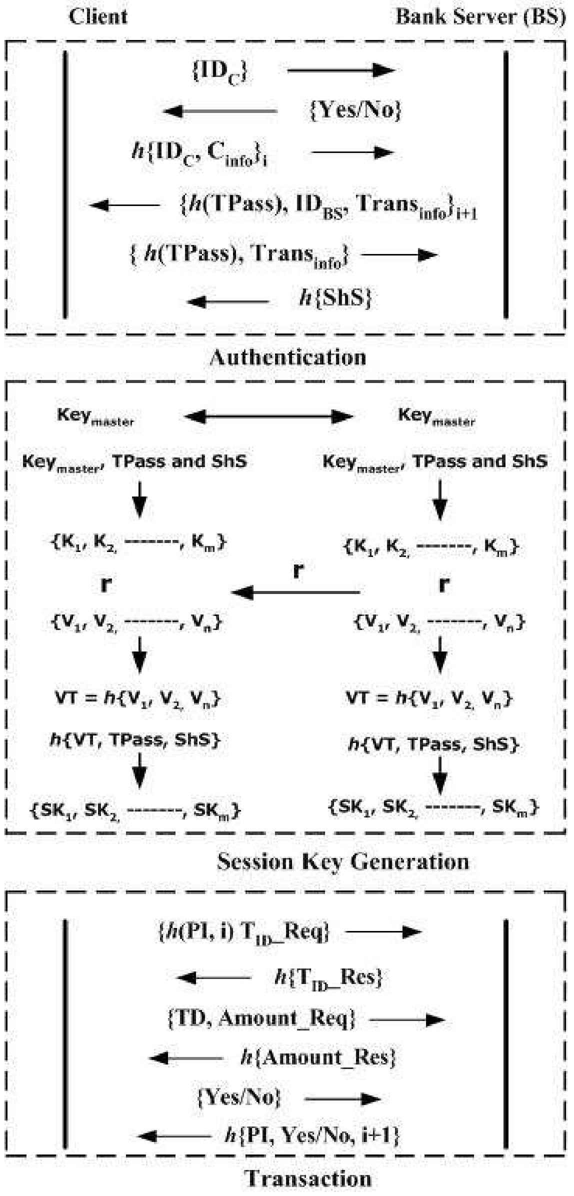

Figure 4.2 illustrates the proposed protocol

A. *User Authentication*

This section details how client C registers with the issuer's server BS and the kind of confidential information which is shared between the two parties. How clients initiate the payment process and the authentication steps that are required is outlined. For clarification it should be noted that the initial registration with BS occurs personally and does not involve any external communication. Therefore, it can be assumed that shared secrets will never be disclosed.

Firstly bank (B) issues a payment card to C, the payment card is in the form of a smart card. Each payment card has a unique ID, which is combined with C's biometric feature (ID_C) to form a Key_{master}.

Key_{master} is a shared key between C and BS where it is manually distributed and stored on the sides of both parties (server and payment card) directly after issuing the payment card. It will not be used in transactions and will only be used for key generation purposes. A TPass will also be issued to clients.

Secondly, C must authenticate themselves to the payment card by providing their biometric feature (e.g. finger print) before each transaction. If authenticated, then the transaction will proceed to the next step. Otherwise access will be denied and the card will be locked. For the transaction process to take place, C needs to initialize an authentication and secret distribution processes with BS. To do so, C sends a hashed message that consists of C's bio ID as well as some related information to BS as follows:

$$\mathbf{C} \rightarrow \mathbf{BS}\text{: } h\{ID_C, C_{info}\}_i \qquad \textbf{(A1)}$$

Upon receipt of the message, BS checks C' bio ID by generating a hashed value and matches it with the sent message. If both hashed values match then BS responds to C's message by sending back BS's ID and requesting C's transaction password and transaction information as shown in (A2). The transaction information includes the type of transaction desired as well as the specific number of key generation steps that is assigned to the transaction. This as we shall see is important in keeping key generation process in pace between the involved parties.

$$\mathbf{BS} \rightarrow \mathbf{C}\text{: } \{h(TPass), ID_{BS}, Trans_{info}\}_{i+1} \ \textbf{(A2)}$$

C replies to BS's request by sending back the required transaction information in addition to the non-long term Transaction Password (TPass) as follows:

$$\mathbf{C} \rightarrow \mathbf{BS} :\{\ h(TPass), Trans_{info}\} \qquad \textbf{(A3)}$$

If TPass is verified then BS sends ShS for identification verification. ShS is a security-enhancing factor as it can be of any value (e.g. C's date of birth), and it is initially used by C to confirm BS's identity as follows:

$$\mathbf{BS} \rightarrow \mathbf{C}\text{: } h\{ShS\} \quad \textbf{(A4)}$$

B. *Session Key Generation*

After confirming their identities, both C and BS perform calculations to generate a set of Secondary Keys (K*i*)s as shown in **(B1-B5)**.

After the completion of generating (K*i*)s, BS generates a random number (r) and sends it to C. (r) could be of any value and used to indicate the start of generating (SK)s process on both sides. It is another security enhancement to confirm the identity of the involved parties. For instance, if one of the involved parties was not genuine and tried to fake (r) and generate (SK)s. Then based on our proposed technique, the fake generated keys will be discovered by the other party as they are not the agreed upon keys and the transaction will not take place. If the identity of BS and C was confirmed the process of generation (SK)s will start as shown in **(B7-B13)**

The number of generated (SK)s is agreed upon between BS and C and is dependant on the number of transactions steps.

Note that the same elimination method in **(B9)** – **(B13)** has been used to clear VT, Pass, and ShS from the system respectively.

Thus, the transaction will start with SK_4. The agreement upon a specific or a limited number of generated (SK)s will determine the number of transactions to be performed between BS and C.

For example, if both parties agreed to generate five (SK)s and suddenly a sixth SK is generated and sent, then based on the proposed scheme, the sixth generated SK will not be recognised and the transaction will be suspended. (SK)s are valid for a short period of time and will not be used again in any other transaction. BS and C can use (SK)s to verify and validate one another's identity by comparing the values of (SK)s, which have already been symmetrically generated. (SK)s are used to validate and verify C and BS to each other during transactions. Only the entire set of (K*i*)s is stored on C's and BS sides, whereas a SK is generated at the beginning of each transaction in order to reduce storage requirements. Moreover, the security of (K*i*)s is guaranteed as they are not transmitted in any transactions.

C. Transaction

In the transaction stage, C starts the transaction process by sending PI, index i and requesting a T_{ID}. BS verifies the contents of the hashed messages by generating other hashed values and matches them with the sent ones. It then responds to C and sends T_{ID}_Res. The index *i* is used to identify the current session key. If the current session key is not the agreed upon key between BS and C then the transaction will fail and the key generation process is required again. If the generated key remains un-known, then the authentication process is required to make sure C is genuine.

Otherwise, C will send the description of the transaction as if it is credit, debit or paying another user in addition to the amount of the payment. BS sends the amount that will be transferred. C can check whether or not the message is the response to their request by comparing the received *h(Amount)* with their requested *Amount*. If they are not matched, C can reject the transaction (No). If matched C then sends back (Yes). If the payment has been approved then BS sends the result back to C with i+1 to indicate the next session key. For more transactions C can use other values in the set of SK_i until being notified to update the secret key. Note that, after each session key has been used, it will be put into all parties' revocation lists in order to prevent the replay of the secrets from both C and BS.

D. Session Key Update

After being used for a specific number of transactions and time period, the set of (SK)s must be updated. To do so, the values of VT **(B9)** also need to be updated. The updating process of a SK is based on the value of the last SK $index_i$. For example, if an update request took place at the end of SK_5 (i = 5) transaction, new values of V_1, $V_2$and V_3 **(B8)** need to be calculated. The three following numbers to 5 have been chosen, these are [6 (i + 1), 7 (i + 2) and 8 (i + 3)].

The values of 6, 7 and 8 can be calculated as follows:

$$\begin{aligned}\text{Value of } 6 &= V_1^{up} = 5\text{mod}(rm)\\ \text{Value of } 7 &= V_2^{up} = 6\text{mod}(rm)\\ \text{Value of } 8 &= V_3^{up} = 7\text{mod}(rm) \quad \textbf{(C1)}\end{aligned}$$

Where

m: is the total number of (K*i*)s generated at the beginning

r: a shared random number between BS and C upon update request.

The updated value of VT will be the hashed value of the updated V_1, V_2 and V_3 values $V_1^{up},V_2^{up},V_3^{up}$ **(C1)** as follows:

$$VT^{up}= h\{ V_1^{up},V_2^{up},V_3^{up} \} \quad \textbf{(C2)}$$

Based on the updated VT [**C2**] we can update session keys as follows:

$$\begin{aligned}SK_1^{up}&= h\{VT^{up}, TPass, ShS\} \quad \textbf{(C3)}\\ SK_2^{up}&= h\{TPass, ShS, SK_1^{up}\} \quad \textbf{(C4)}\\ SK_3^{up}&= h\{ShS, SK_1^{up}, SK_2^{up}\} \quad \textbf{(C5)}\\ SK_4^{up}&= h\{SK_1^{up}, SK_2^{up}, SK_3^{up}\} \quad \textbf{(C6)}\\ SK_5^{up}&= h\{SK_{(n-1)}^{up}, SK_{2(n-2)}^{up}, SK_{(n-3)}^{up}\} \quad \textbf{(C7)}\end{aligned}$$

Note that both VT and ShS can be updated at the same time. So, to avoid using the same values for updating ShS, we ignore [(i + 1), (i + 2) and (i + 3)] from previous SK update and use i + 4. We then multiply i + 4 with the value of the old shared secret as shown below:

$$ShS^{up} = ShS * i + 4$$

TPass and ShS can be both changed or updated by C over the Internet or in person. Key_{master} also has to be updated although it is considered as a long term key. Biometric features are prone to changes due to age, illness or injury. Therefore, a regular update is required. Privacy and convenience of clients should be a paramount when doing so, to minimise the usual discomfort associated with the process. Clients should also keep the issuer informed if any changes that would either stop them or make it difficult for them to use their biometric features have occurred. So other alternatives can be arranged.

E. Failure Recovery

Network failures occur more frequently due to low network reliability. It is understood that failure recovery is relevant to key synchronization process, in that; there should be an efficient key synchronization mechanism. Therefore, the proposed scheme is able to deal with a failure situation when it occurs, especially when concurrent sessions are being performed.

For instance, six sessions (SK_1–SK_6) are performing concurrently and have been used. If SK_4 fails, SC can restart SK_4 with SK_n, where SK_n can be any value in the set of the generated (SK)s other than SK_1-SK_4, and will be recognized by BS (i.e. SK_5, SK_6). The same applies when a session key fails after reaching BS, C can also do the same.

Note that (SK)s used before in transaction can not be used again for obvious reasons, as long as (SK)n used in recovery does not exceed the number of (SK)s agreed upon between BS and SC, the transaction would continue normally. If SK_6 fails, generation of (SK)s should start from the beginning. Having the set of (Ki)s stored on the device, it saves the inconvenience of starting the authentication process over again. Also it prevents redistribution of shared factors, which minimize the risk of attacks.

V. Discussion

In the proposed technique, generating each session key is not based only on the master key. Thus, the compromise of the master key will never compromise the security of the system. In this section, we discuss about the security of the proposed technique.

A. Security of Dynamic Key

This proposed technique never reprocesses a SK during a transaction. If the fraudulent user succeeds in generating some (SK)s and is trying to guess the next SK, BS can keep track of the total number of incorrectly hashed or encrypted messages. When the number of incorrect messages exceeds a predetermined limit (the number of SKs agreed between C and BS), BS can then delete the registered C. To reactivate the service, C must to start with new set of (Ki)s, which means the fraudulent user has to start the process of collecting the required keys again and obtaining unique IDs from the beginning to generate the SKs. The fraudulent users must also be able to authenticate themselves to the SC and to generate (SK)s based on the genuine (r), which is just known to BS and C:

$$\mathbf{BS} \rightarrow \mathbf{C}\text{: } \{\text{Random(r)}\}_{\text{i}} \quad (1)$$
$$\mathbf{C} \rightarrow \mathbf{BS}\text{: } \{\text{ Random(r)}\}_{\text{i+1}} \quad (2)$$

It is also difficult to retrieve the generated SK, even assuming the fraudulent user can intercept the message and successfully retrieves the key. It is not feasible to obtain the three secure factors and work out the technique that is been used to generate the (SK)s:

$$N_{BS} = \mathbf{BS} \rightarrow \mathbf{C}\text{: } \{\text{Random(r)}\}i$$

$$C \mid\equiv BS \overset{(i)}{\Leftrightarrow} C, C \lhd (X)$$

C believes BS has sent the message X. (i) is a secret that is only known to C and BS. C is able to see it.

$$N_{C1} = \mathbf{C} \rightarrow \mathbf{BS}\text{: } \{\text{Random(r) }\}i+1$$

C also could send (r) to BS.

$$BS \mid\equiv C \overset{(i+1)}{\Leftrightarrow} BS, BS \lhd (X)$$

BS believes C has sent the message X where (i+1) is a secret that is only known to C and BS. BS is able to see it.

It shows that only BS and C have the authority to reveal the exchanged information.

$$N_{C2} = \mathbf{C} \rightarrow \mathbf{BS}\text{: } h\{\text{CToken}\}_{SKi}$$

BS sees **C'**s generated Token.

$$(BS \lhd \text{ Token})$$

BS believes the message sent by C.

$$(BS \mid\equiv C \mid\sim N_{C2})$$
$$BS \rightarrow C\text{: } \{\text{Yes/No}\}$$

In the proposed technique, the number of (SK)s used is always one more than the number of data retrievals that the member wishes to perform. In the worst case scenario, if the fraudulent user has guessed all the correct values of the (SK)s, and BS has failed to track the fake messages, then the short life span of (SK)s will overcome the threat once the time period of the current set of SKs has run out, the compromised (SK)s are no longer valid and a new set of (SK)s are generated by the valid parties.

By applying the efficient key generation technique, the fraudulent values evolved can still be detected by BS when it receives a hashed message with an old (SK).

B. Implementation Issues

In this section, we discuss major issues related to the implementation of our scheme in Internet banking.

1) Keys Distribution, Storing and Managing

Our technique focuses on the deployment of the Key_{master} rather than the semi-secret credit-card number. Therefore, Key_{master} needs to be distributed between C and BS. Before making the first payment, C needs to register with B in order to share Key_{master}. In the case that C is provided a smart card, B can generate Key_{master} and store it on the card before issuing the card to C. Therefore Key_{master} does not need to be transferred in any transaction which results in fully offline key generation.

The proposed technique requires two sets of keys, K_i and (SK)s, to be generated at each party's device. However, only the entire set of preference keys (Ki)s is stored on both parties devices, whereas each member in the set of session keys (SK)s is generated at the beginning of each transaction in order to reduce storage requirement. The security of (K_i)s is guaranteed as they are not transmitted in any transaction. After generating the set of (K_i)s, the master key is no longer used in the system. As well as the session key SK, after a new session key has been generated, the previously used key is then removed from the system.

C. Security of the Proposed Technique against Attacks

Consider the situation where a (SK)i is transmitted in clear text over an unsecured channel. First of all, due to the one-way property of the hash function, reverse operation of (SK)i to retrieve (Ki)s is computationally unfeasible. A one-way hash function is preferred due to its proven security.

There is a possibility of collecting a number of (SKs) and trying to guess the next value of (SK)i. However, the fraud will be detected and limited by allowing a limited number of attempts for specific C.

If the attempts exceed the specified limit, C's account is suspended. The system then notifies C that there were unauthorized attempts on their account and asks for SKs update. After C has updated SKs, the fraudulent user must repeat the attacking processes from the beginning. Also consider that these attempts have not been detected by the system. The set of (SK)s is valid over this short period of time. After the session keys are updated, the keys used by the fraudulent user are no longer valid.

As the initial input for the set of SKs are (Ki)s and V values, the fraudulent user must be able to record all (SK)s from SK_1. They must then try to re-compute (Ki)s and V values. In the case that the attempt is successful, the fraudulent user can generate the next SK, which could be used in fraud. Consequently, C does not need to update the Key_{master}.

Instead C asks the BS to update the SKs set. After the new set of SKs becomes valid, the current (SK)s will no longer be valid as they rely on the (Ki)s that have not been used in any transaction. Therefore, to retrieve (Ki)s, the fraudulent user needs to capture the transaction with SK_1 and attempt to compute (Ki).

The only possible successful attack to the proposed technique is that the fraudulent user must be able to do the following:

1 Access each party's device to retrieve the entire set of (Ki)s,
2 Record all SKs transmitted in all transactions, and
3 Detect the request to update the set of SKs.

In the worst case scenario, if the fraudulent user succeeded in the above process, they can generate and use the valid SKi until being detected by the system. Generating each set of (SK)s is based on dynamic parameters randomly chosen from the set of (Ki)s. As a result, the higher number the transactions are performed, the less the chance that the system will be compromised.

C. Security Analysis

In this section, we show that our new scheme has advantages over SET [14] and iKP [26] in terms of payment transaction security when applied to Internet banking. Based on the payment model described in section IV, the transferred messages in the proposed protocol deliver the same information and purposes as that of SET and iKP.

Our protocol also satisfies a non-repudiation feature that is known as one of the important security properties of both SET and iKP. Every party is capable to deliver non-repudiable evidence to the other party in case of illegal attempts.

Table 1 demonstrates our scheme security enhancements over SET [14] and iKP [26] based on number of cryptographic operations:

Table 1
Illustrates the numbers of cryptographic operations by SET, iKP, and our work respectively

Cryptographic Operations		SET	iKP	Ours
1. Public-key encryptions	C	1	1	-
	BS	1	-	-
2. Public-key decryptions	C	-	-	-
	BS	2	1	-
3. Signature generations	C	1	1	-
	BS	1	1	-
4. Signature verifications	C	2	3	-
	BS	1	2	-
5. Symmetric-key encryptions/decryptions	C	2	-	4
	BS	1	-	2
6. Hash functions	C	3	2	3
	BS	-	1	4
7. Keyed-hash functions	C	-	-	2
	BS	-	-	1
8. Key generations	C	-	-	2
	BS	-	-	1

It is not hard to see that in the proposed protocol, we only apply symmetric-key operations including MAC and hash functions that lead to higher level of security than SET and iKP. In comparison with SET and iKP, the client is required to perform both public key encryptions and signature verifications which lead to more computational tasks and less security protection.

Even though, the key generation process requires a regular keys update in our technique. However, this would not cause security issue as the key generation processes can be done offline.

Based on the comparison with the existing payment systems, our proposed scheme has the following advantages:

- Facilitates fraudulent payments prevention by applying different types of security mechanisms.
- Incorporates a DKG to prevent access by fraudulent users by confirming that involved parties can meet the secret keys generation requirements and are allowed to perform transactions
- Reduces fraudulent attempts due to strong identity verification process that captures biometrics
- Combines both biometrics and smart card to make it extremely secure and provide excellent user-to-card authentication.
- Is able to adapt to changes with future technology

VI. Conclusion

In this paper, we have proposed an efficient new scheme which can prevent fraud by applying different security algorithms, generating and updating limited-use secret keys. It uses advanced authentication technologies and is well adapted to any possible future technology. Moreover, it does not rely on fixed values where hacking one secret will not compromise the whole system's security. The generation of each set of keys is based on dynamically generated preference keys. The higher number the transactions performed, the less chance the system has of being compromised. The practical usefulness of the technique has been demonstrated by applying it to Internet banking payment systems. The results show that our technique enhances their security considerably.

It has been shown that the proposed technique is secure against key compromise. For future work, we aim to analyze the security of the system that applies the proposed technique. Moreover, we aim to apply the proposed technique to other kinds of internet applications, especially mobile commerce.

References

[1] A.O. Freier, P. Karlton, and P. Kocher. The SSL Protocol Version 3.0: Internet Draft, March 1996. http://wp.netscape.com/eng/ssl3/ssl-toc.html.

[2] A. Shamir. Secureclick: A web payment system with disposable credit card numbers. In Proceedings of Financial Cryptography, pages 232–242, 2001

[3] Bonchi, F. Giannotti F., Mainetto G., Pedreschi D., "A classification-based methodology for planning audit strategies in fraud detection", August 1999 Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining

[4] Chan, P.K.; Fan, W.; Prodromidis, A.L.; Stolfo, S. J, "Distributed data mining in credit card fraud detection", Intelligent Systems, IEEE [see also IEEE Expert],Volume: 14 Issue: 6, pp.67 -74, Nov.-Dec. 1999

[5] C. Corzo, F. Corzo S., N. Zhang and A. Carpenter, 2006 Using Automated Banking Certificates to Detect Unauthorised Financial Transactions, Computer Science Volume 4107/2006,

[6] Cooke, J.C.; Brewster, R.L.; 1993 " The use of smart cards in personal communication systems security" Telecommunications, 1993. Fourth IEE Conference on 18-21 Apr 1993 Page(s):246 - 251

[7] Cristian Radu, 2003: Implementing Electronic Card Payment Systems (Artech House Computer Security Series creditcards.com, 2006- http://www.creditcards.com/history-of-credit-cards.php

[8] David S. Evans, Richard Schmalensee, 2003; Paying with Plastic: The Digital Revolution in Buying and Borrowing (Second Edition)

[9] Donal O'Mahony, Michael Peirce, and Hitesh Tewari. Electronic Payment Systems for E-Commerce. Artech House, 2001. Second edition.

[10] Dugelay, J.-L.; Junqua, J.-C.; Kotropoulos, C.; Kuhn, R.; Perronnin, F.; Pitas, I.; Recent advances in biometric person authentication, 2002, [online], Available, IEEE Xplore, Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASSP '02). IEEE International Conference on , Volume: 4 , 13-17 May 2002 Pages:IV-4060 - IV-4063 vol.4, http://ieeexplore.ieee.org

[11] Elliot, S.; Loebbecke, C.; 1998 " Smart-card based electronic commerce: characteristics and roles" System Sciences, 1998., Proceedings of the Thirty-First Hawaii International Conference on Digital Object Identifier

[12] E. V. Herreweghen, Non-Repudiation in SET: Open Issues, LNCS Vol. 1962, 2001, pp. 140-156

[13] Gennady Medvinsky and Clifford Neuman. NetCash: A design for practical electronic currency on the Internet. In Proceedings of the 1st ACM Conference on Computer and Communications Security, pages 102{106, November 1993

[14] H. Krawczyk. Blinding of credit card numbers in the SET protocol. Lecture Notes in Computer Science, 1648:17{28, 1999

[15] Hugo Krawczyk. The Order of Encryption and Authentication for Protecting Communications (or: How Secure is SSL?). In Joe Kilian, editor, Advances in Cryptology { CRYPTO 2001, Lecture Notes in Computer Science, LNCS 2139, pages 310{331. Springer-Verlag, August 2001

[16] John G. Faughnan, "International Net-Based Credit Card/Check Card Fraud with Small Charges". http://www.faughnan.com/ccfraud.html#CPBank

[17] Jon M. Peha, Ildar M. Khamitov, "PayCash: a secure efficient Internet payment system", Proceedings of the 5th international conference on Electronic commerce, Pittsburgh, Pennsylvania, pp.125-130, September 2003.

[18] Joris Claessens, Valentin Dem, Danny De Cock, Bart Preneel, and Joos Vandewalle. On the Security of Today's Online Electronic Banking Systems. Computers & Security, 21(3):253{265, June 2002.

[19] K. Dehnad, "A Simple Way of Improving the Login Security, Computers & Security, vol. 8, pp. 607-611, 1989.

[20] Lars Rasmusson and Sverker Jansson. Simulated Social Control for Secure Internet Commerce. In Proceedings of the 1996 ACM workshop on New Security Paradigms, pages 18{25, September 1996.

[21] Medvinsky, G. & Neuman, B. C. (1993). Netcash: A design for practical electronic currency on the internet. Proceedings Of First ACM Conference On Computer and Communication Security, ACM.

[22] N. Asokan, P. Janson, M. Steiner and M. Waidner, "The State of the Art in Electronic Payment Systems", in IEEE Computer, September 1997.

[23] N. Ahituv, Lapid, Y., Neumann, S., Verifying the Authentication of an Information System User," Computers & Security, vol. 6, pp. 152- 157, 1987.

[24] Patiwat Panurach, "Money in electronic commerce: digital cash, electronic fund transfer, and Ecash", Communications of the ACM, Volume 39, Issue 6, pp.45-50, 1996

[25] Paul Ashley, Mark Vandenwauver, and Joris Claessens. A Comparison of SESAME and SSL for Intranet and Internet Security. In Jan HP Eloff and Rossouw von Solms, editors, Information Security { Small Systems Security & Information Security Management { Proceedings of the sixth working conference of IFIP WG 11.1 & 11.2, pages 60{69. Kluwer Academic Publishers, September 1998.

[26] R. Hauser, M. Steiner, and M. Waidner. Micro-payments based on iKP. IBM Research Report RZ 2791, IBM Research Division, December 1996.

[27] Rubin, A.D., Wright, R.N., Off-Line Generation of Limited-Use Credit Card Numbers. LNCS, Vol. 2339 (2002) 196-209

[28] Sherif, M.H.; Serhrouchni, A.; Gaid, A.Y.; Farazmandnia, F.; 1998 SET and SSL: electronic payments on the Internet, Computers and Communications, 1998. ISCC '98. Proceedings. Third IEEE Symposium on 30 June-2 July 1998 Page(s):353 – 358

[29] S. Kungpisdan, P.D. Le, and B. Srinivasan. A limited-used key generation scheme for internet transactions. Lecture Notes in Computer Science, 3325:302{316, 2005

[30] Steven M. Bellovin. Cryptography and the Internet. In Hugo Krawczyk, editor, Advances in Cryptology { CRYPTO'98, Lecture Notes in Computer Science, LNCS 1462, pages 46{55. Springer-Verlag, August 1998.

[31] W.-K.; Liew, C.-C.; Ng, Lim, E.-P.; Tan, B.-S.; Ong, K.-L.; 1999 Non-repudiation in an agent-based electronic commerce system, Database and Expert Systems Applications, 1999. Proceedings. Tenth International Workshop on 1-3 Sept. 1999 Page(s):864 - 868
Digital Object Identifier 10.1109/DEXA.1999.795295

Osama Dandash is a phd student at School of Information Technology. Osama's research area is Internet Banking Payment Security. He holds two masters in network engineering and network computing from RMIT and Monash Universities. Besides studying, Osama works as a sessional lecturer at Monash University.

Previous publications include:

- **"Wireless Internet Payment System Using Smart Cards"**, the IEEE International Conference on Information Technology, pp. 16-21, Arp 2005
- **“A new Dynamic Key Generation Scheme for Fraudulent Internet Payment Prevention”**, Information Technology, 2007. ITNG '07. Fourth International Conference on 2-4 April 2007 Page(s)
- **“A new Group Key Management Structure for Fraudulent Internet Banking Payments Detection”**, ICEIS 2007, 9th International Conference on Enterprise Information Systems 12-16, June 2007, Funchal, Madeira – Portugal,

Yiling Wang is a phd student at School of Information Technology. Yiling's research area is Group Key Management in Wireless networks. Yiling holds a master in network computing from Monash University.

Previous publications include:

- **“Efficient Group Key Management in Wireless Networks”**, IEEE Proceedings of Information Technology: New Generations, pp. 432-437, USA 2006.
- **“Hybrid Group Key Management Scheme for Secure Wireless Multicast”**. 6th IEEE International Conference on Computer and Information Science (ICIS 2007). pp: 346-352
- **“Scalable multi-subgroup key management in wireless networks”**. International Journal of computer science and network security. pp:95-106

Dr Phu Dung Le is currently working at School of Information Technology. Dr Le's main research interests are:

- Image and Video Quality Measure and Compression
- Intelligent Mobile Agents
- Security in Quantum Computing Age

He used to teach Data Communication, Operating System, Computer Architecture, Information Retrieval and Unix Programming. He has also researched in Mobile Computing, Distributed Migration. Currently he is lecturing network security and advanced network security in addition to supervising phd students.

Previous publications include:

- **"A Tool for Migration to Support Resource and Load Sharing in Heterogeneous Environments"**, Proceedings of the International Conference on Networks, pp. 83-87, Feb 1996
- **"A Limited-used Key Generation Scheme for Internet Transactions"**. Information Security Applications, Vol. 3325, pp 302-316, Lecture Notes in Computer Science, ISBN: 3-540-24015-2, Korea, 2005
- **“The Design and Implementation of a Smart Phone Payment System”**, IEEE Proceedings of Information Technology: New Generations, pp. 458-463, USA 2006