A Proposed Electronic Consumer Protection Model for Enhancing E-commerce Services in Egypt

By

Mohamed Saad Ahmed
Computer and Information System Department,
Sadat Academy for Management Science,
Cairo, Egypt
mohamedsa3dfayed@gmail.com

Prof. Dr. Farahat Farag Frahat,
Computer and Information System Department,
Sadat Academy for Management Science,
Cairo, Egypt
Fararhatfarag2021@gmail.com

Abstract

E-commerce refers to businesses and consumers buying and selling products online. The preponderance of e-commerce websites on the internet are retail stores selling products directly to the public. While buying products online, it is very important that you select the right payment method for yourself. Before you buy an item, always make sure that the seller's payment methods will work for you. The Researcher represent new method is the best and recommended online payment method. Wallet method has a reputation for security, protecting the interests of both merchants and customers. It is also a convenient option for patrons and merchants. Consumers spend less time entering their information; merchants can set up a payment system quickly, with no upfront payment necessary. An e-commerce payment system facilitates the acceptance of electronic payment for online transactions

Keywords: E-commerce, Online payment, Encrypted Algorithms, MD5, Checksum.

1-Introduction

The world saw many developments in many fields in recent years. One of the manifestations of this development is the revolution in communications. This development surpasses the industrial revolution, loading to the emergence of the international internet network, which contributed to the development of international trade.

This recent scientific and technological development has allowed many contracts to be concluded quickly and easily among all people across the world. It was difficult before this time. The consumers are transformed from simple natural products to complex and dangerous new forms of products [1].

One of the most serious topics in this field is the contracts. This is a new form of contracts, which are without known methods between two parties. One of them is present and the other is absent. All of them are linked through modern means of communication invented by the human, which has become the world's entire parties, short distances, and spatial, temporal barriers. they allowed the multiplication of risks to consumers, forced states to work on framing these transactions in terms of legislation to control them and mitigate the risks resulting from them [2].

Because the electronic transactions mentioned above are the weak side of the consumer, this makes them vulnerable to manipulation of risks. Also, the power and economic dominance of professionals, which requires lawmakers to enact special laws or improve existing laws to rebalance contractual relationships between consumers and professionals.

2. Related Works

There were some papers and professional articles on the general topic of E-commerce in different areas and Researchers had a belief that the growth of E-commerce depended on many security- related factors [3-6] such as:

2.1.1 Benefits of E-Commerce and it's security " this problem was studies by

Customer Relationship Management system facilitates investing in customer service, getting personal, having variety, helping the customer, analyzing and using old business rules and bringing everything together in one system [7].

2.1.2 How to increase consumers' trust in E-Commerce " this problem was studies by

A clear example of escalating security to increase trust derives from people being willing to deal with E-commerce if they feel assured that their credit card numbers and personal data are protected through cryptography. Therefore, PKI (Public Key Infrastructure) was treated as a solution to E-commerce security and privacy concerns [8]. 2.1.3 Legal frameworks requirement of E-Commerce" this **problem was studies by**

The challenges that consumers faced have aroused the need to adapt existing legal and regulatory frameworks to the particular requirements of E-commerce. Even the United Nations guidelines for consumer protection have been revised due to the change which has occurred in the current environment for both consumers and businesses [9].

2.1.4 Consumer low and Policy " this problem was studies by.

E-commerce with all its developments and challenges because of its continuous growth drove the Intergovernmental Group of Experts on Consumer Protection Law and Policy, at its first session, held on 17 and 18 October 2016, to demand the UNCTAD (United Nations Conference on Trade and Development) secretariat to prepare a report on E-commerce for consideration at its second session [11].

3. Methodology

In this research, the researcher will rely on the descriptive approach to familiarity with the subject of the study and provide details on its most important aspects. This is done by using case study method for applying the protection model for enhancing E-commerce services in Egypt website for online shopping.

The main objective of this study is to Develop a proposed Electronic Consumer Protection Model for Enhancing E-commerce Services in Egypt This model will minimize the risks in an E- commerce environment like cheating, fraud, circumvention, bowing and compliance.

The study moved on certain phases described in the framework of the experimentation as shown in Fig.1. The framework includes many phases: Customer / Buyers, building E-commerce website, Customer Sign Up as E-Commerce Website, Trader place order and proceed payment, apply encrypted Algorithm (MD5 and checksum), and Seller Received payment and process order.

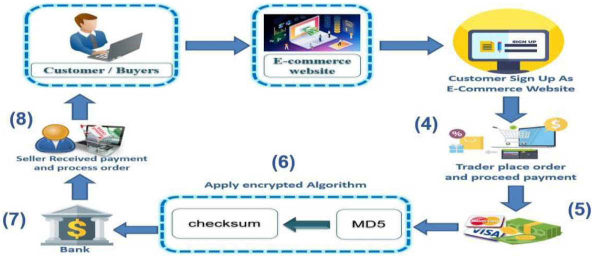

Customer / Buyers
E-commerce website
Customer Sign Up As E-Commerce Website
(4)
Trader place order and proceed payment
(5)
MD5
checksum
(6)
Apply encrypted Algorithm
(7)
Bank
(8)
Seller Received payment and process order

Fig. 1. The conceptual Model for process of sending a payment to Wallety Checkout

3.1 Md5 algorithm

The MD5 message-digest algorithm is a widely used hash function producing a 128- bit hash value. Although MD5 was initially designed to be used as a cryptographic hash function, it has been found to suffer from extensive vulnerabilities. It can still be used as a checksum to verify data integrity, but only against unintentional corruption. It remains suitable for other non- cryptographic purposes, for example for determining the partition for a particular key in a partitioned database. MD5 was designed by Ronald Rivest in 1991 to replace an earlier hash function MD4, and was specified in 1992 as RFC 1321. One basic requirement of any cryptographic hash function is that it should be computationally infeasible to find two distinct messages that hash to the same value. MD5 fails this requirement catastrophically; such collisions can be found in seconds on an ordinary home computer.

The weaknesses of MD5 have been exploited in the field, most infamously by the Flame malware in 2012. The CMU Software Engineering Institute considers MD5 essentially "cryptographically broken and unsuitable for further use". As of 2019, MD5 continues to be widely used, in spite of its well-documented weaknesses and deprecation by security experts.

One of the most widely used Cryptographic hash Function is MD5 or" message digest 5". MD5 creates a 128-bit message digest from the data input which is typically expressed in 32 digits hexadecimal number. MD5 hashes are unique for different inputs regardless of the size of the input. MD5 hashes looks like this

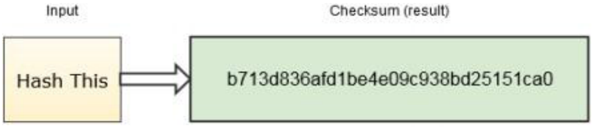


Fig. 2: Hexadecimal representation of input by md5

It is widely used to make sure that the transferred file in a software has arrived safely. For example, when you download a file from the Internet/Server it might be corrupted or there might be data loss due to connection loss, virus, hack attack or some other reason. One way to check if the downloaded file is same as you intended is by generating an MD5 hash on the server for the file and again for the downloaded file, if both of the hash matches then your file is perfect. It is also used in database to store passwords as hash instead of the original input.

3.1.1 Processing the blocks MD5

The contents of the four buffers (A, B, C and D) are now mixed with the words of the input, using the four auxiliary functions (F, G, H and I). There are four rounds, each involves 16 basic operations. One operation is illustrated in the figure below.

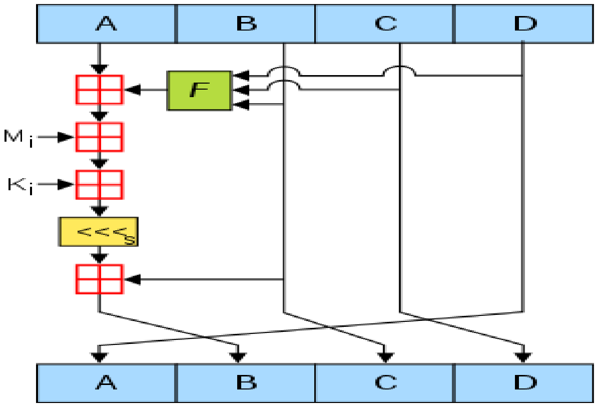


Fig. 3. One MD5 operation. MD5 consists of 64 of these operations, grouped in four rounds of 16 operations. F is a nonlinear function; one function is used in each round. Mi denotes a 32-bit block of the message input, and Ki denotes a 32-bit constant, different for each operation.

The figure shows how the auxiliary function F is applied to the four buffers (A, B, C and D), using message word Mi and constant Ki. The item "<<<s" denotes a binary left shift by s bits.

3.2 Secure Hash Function Algorithm (SHA)

Family of SHA comprise of four SHA algorithms; SHA-0, SHA-1, SHA-2, and SHA-3.
Though from same family, there are structurally different.

SHA-1 or Secure Hash Algorithm 1 is a cryptographic hash function which takes an input and produces a 160-bit (20-byte) hash value. This hash value is known as a message digest. This message digest is usually then rendered as a hexadecimal number which is 40 digits long. It is a U.S. Federal Information Processing Standard and was designed by the United States National Security Agency.
SHA-1 is now considered insecure since 2005. Major tech giants' browsers like Microsoft, Google, Apple and Mozilla have stopped accepting SHA-1 SSL certificates by 2017.

To calculate cryptographic hashing value in Java, MessageDigest Class is used, under the package java.security.

3.2.1 What is hashing algorithm? How it works?

As we discussed, a hash function lies at the heart of a hashing algorithm. But, to get the hash value of a pre-set length, you first need to divide the input data into fixed sized blocks. This is because a hash function takes in data at a fixed-length. These blocks are called 'data blocks.' This is demonstrated in the image below.

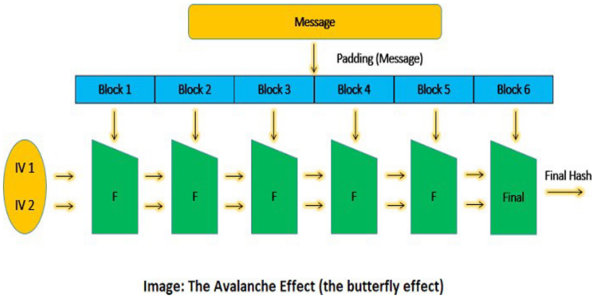


Fig. 4 Hashing Algorithms Work

3.3 Wallety payment Method with Merchant Integration

Wallety enables merchants to use payment enabled websites, e-commerce or other applications by providing a low effort integration solution. It is suitable for most website hosting environments as merchants can integrate payment capabilities into their application without installing or configuring any payments software.

Our 3-Party Payment model (the merchant, Wallety, and the cardholder) allows Wallety Payment Server to manage the payment pages and collect the cardholder's card details on your behalf. The advantage of 3-Party payments is that the complexity of securely collecting and processing card details is handled by the Wallety Payment Server, allowing you to focus on your website part of the payment process.

The cardholder's Internet browser is redirected to take the Transaction Request to the Wallety Payment

Server to process the transaction. After processing the transaction, the cardholder's Internet browser is returned to a web page that you nominate in the transaction together with a Transaction Response. The Transaction Response processing of the receipt information finalizes the transaction. The cardholder's browser provides the redirect method to communicate he information between the merchant and Wallety.

This is an asynchronous connection and the cardholder leaves your web site to go to the Wallety Payment Server, which means the transaction is broken or disrupted into 2 distinct sessions, the creation of the Transaction Request and the processing of the Transaction Response.

3.3.1 About Merchant IDs

To use Merchant Administration, a merchant profile is required. The profile is a record of merchant details and the permitted functionality that the merchant has within their portal; all details are stored on the Wallety Payment Server.

Two types of merchant profile are created through the bank's enrolment process:

TEST Merchant Profile (Draft Profile)-this allows merchants, within the test facility, to perform transactions against an emulator of Wallety's transaction processing system.

In TEST mode only, the decimal value will determine the Acquirer/Issuer Response Code (eg, 5995 will return an Acquirer Response Code of 95). Once you move to production the response will be provided by the Issuer and has nothing to do with the Amount's decimal value. Please use an Amount that is a multiple of 100 to simulate an approved transaction during TEST mode (eg, 100, 43500, 700).

PRODUCTION Merchant Profile-activates merchants within the production system, allowing them to process transactions directly against the Wallety live transaction processing system. This profile is only activated once testing has been deemed sufficient by the bank.

Must to have Payment Service Provider (PSP) when you have completed your testing and have E- Commerce Website. The Production merchant ID and gateway ID must be enabled by the PSP first.

3.3.2 Prerequisites

This section lists the requirements and basic steps you need to take to build a successful integration.

Support Material and Information, you must have the following:

1. Merchant Profile Access
2. This is delivered as soon as the bank enrolls you into Wallety
3. Gateway access codes
4. Are available through your gateway section once you login
5. Test Customer account
6. Will be provided upon requests

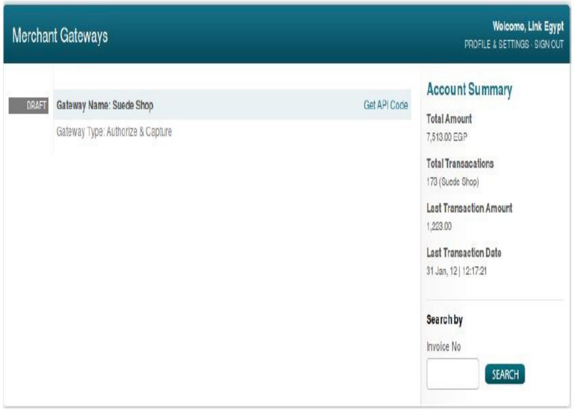
Note that the initial setup of any gateway on the system will be in Test Mode (Draft Mode) until the bank verifies that this gateway is ready for production and the bank decides to switch this merchant's gateway to Live Mode. (a label on every gateway will present the current state of the gateway) Whilst in Draft Mode the merchant account will be given test Wallety Customer accounts with test Customer Payment Profiles, to test his integration.

3.3.3 Story Board for Wallety Checkout Portal

1. The user will hit a Wallety button from the merchant's website posting data into the Wallety Checkout Portal.
2. If the merchant data is valid, the Customer will then need to login with his Wallety Customer Account (or else he will need to create an account, create a payment profile and validate in three simple steps)
3. The Customer now will need to complete a few fields (2-3 fields) to hit the checkout.
4. The transaction is processed, and upon response a result is given to the user and the user is automatically redirected to the URL of your designation with the results of the transaction in the URL.

3.3.4 Wallety Checkout Gateway Integration

A merchant with access to Wallety Merchant Portal can now access his gateways panel as shown in figure 5.



Merchant Gateways

Welcome, Link Egypt

PROFILE & SETTINGS · SIGN OUT

DRAFT Gateway Name: Suede Shop

Gateway Type: Authorize & Capture

Get API Code

Account Summary

Total Amount
7,513.00 EGP

Total Transacations
173 (Suede Shop)

Last Transaction Amount
1,223.00

Last Transaction Date
31 Jan, 12 | 12:17:21

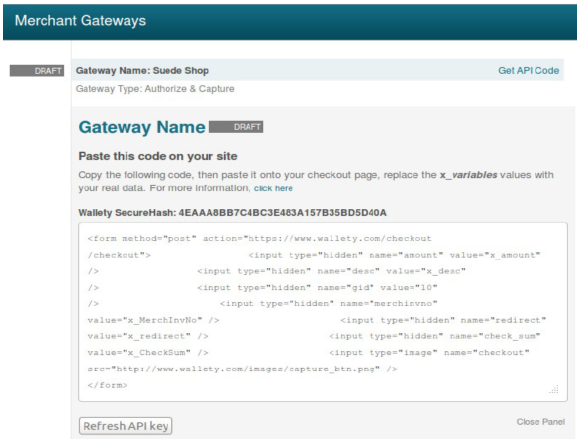
Search by

Invoice No

SEARCH

Fig. 5. Access to Wallety Merchant Portal

The access credentials and an (API code) sample are left for the merchant per gateway for ease of use, here's a sample of such a screen as shown in figure 6.



Merchant Gateways

DRAFT Gateway Name: Suede Shop

Gateway Type: Authorize & Capture

Get API Code

Gateway Name DRAFT

Paste this code on your site

Copy the following code, then paste it onto your checkout page, replace the ***x_variables*** values with your real data. For more information, click here

Wallety SecureHash: 4EAAA8BB7C4BC3E483A157B35BD5D40A

```
<form method="post" action="https://www.wallety.com/checkout
/checkout">                    <input type="hidden" name="amount" value="x_amount"
/>                  <input type="hidden" name="desc" value="x_desc"
/>                  <input type="hidden" name="gid" value="10"
/>                      <input type="hidden" name="merchinvno"
value="x_MerchInvNo" />                     <input type="hidden" name="redirect"
value="x_redirect" />                     <input type="hidden" name="check_sum"
value="x_CheckSum" />                     <input type="image" name="checkout"
src="http://www.wallety.com/images/capture_btn.png" />
</form>
```

Refresh API key

Close Panel

Fig. 6. Access to Wallety Merchant Portal

For the merchant to setup his website to process transactions through Wallety, the merchant should add this snippet of code at the point he wishes his consumer should proceed to pay.

```
<form method="post" action="https://www.wallety.com/checkout/checkout">

<input type="hidden" name="amount" value="x_amount" />

<input type="hidden" name="desc" value="x_desc" />

<input type="hidden" name="gid" value="10" />

<input type="hidden" name="merchinvno" value="x_MerchInvNo" />

<input type="hidden" name="redirect" value="x_redirect" />

<input type="hidden" name="check_sum" value="x_CheckSum" />

<input type="image" name="checkout" src="https://www.wallety.com/images/ capture_btn.png" />

</form>
```

As mentioned previously the above snippet is an example of what is to be expect. Wallety Checkout Portal only accepts requests through HTTP POST requests (You see the snippet of code: form method is post) using http or https.

3.3.5 The process of sending a payment to Wallety Checkout Portal, is through the following steps.

1. Fill in all mandatory fields in the above form with appropriate values.

2. Provide a fingerprint/check sum through the following algorithm

   a. ASCII sort all fields by field name

   b. adds all fields together as string

   c. adds the Wallety Secure Hash (you can know it from your Wallety Merchant Portal gateways panel) at the beginning of the output string of step 2

   d. encrypts your result with MD5

   e. converts your encrypted result characters to uppercase

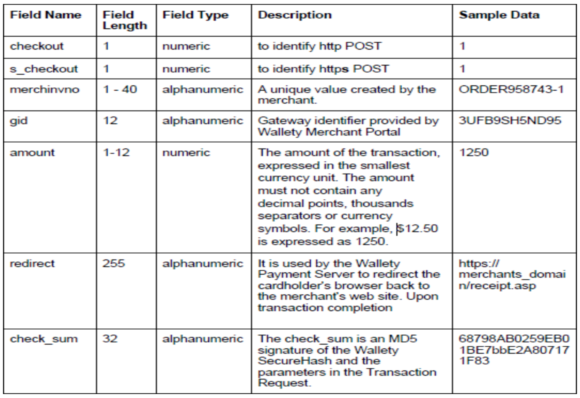
   f. Place the value in the form in the checksum field

3. Post form to Wallety Checkout Portal.

4. Expect the results on the redirect URL you issued.

Here is a table (1) describing most of the field types to be expected additional notes are at the end of each table.

Table 1 The Process of Sending a Payment to Wallety



Field Name	Field Length	Field Type	Description	Sample Data
checkout	1	numeric	to identify http POST	1
s_checkout	1	numeric	to identify https POST	1
merchinvno	1 - 40	alphanumeric	A unique value created by the merchant.	ORDER958743-1
gid	12	alphanumeric	Gateway identifier provided by Wallety Merchant Portal	3UFB9SH5ND95
amount	1-12	numeric	The amount of the transaction, expressed in the smallest currency unit. The amount must not contain any decimal points, thousands separators or currency symbols. For example, $12.50 is expressed as 1250.	1250
redirect	255	alphanumeric	It is used by the Wallety Payment Server to redirect the cardholder's browser back to the merchant's web site. Upon transaction completion	https://merchants_domain/receipt.asp
check_sum	32	alphanumeric	The check_sum is an MD5 signature of the Wallety SecureHash and the parameters in the Transaction Request.	68798AB0259EB01BE7bbE2A807171F83

3.3.6 Description of the input Fields

3.3.6.1 checkout or s_checkout:

You need to set at least one of those fields with 1 for Wallety to start processing your call. If both fields are present the server will prefer s_checkout and neglect checkout, meaning it will neglect any unsecure post if s_checkout is present at any time.

Note: Checkout defines that your POST was sent from an http page whilst s_checkout denotes that your POST was sent from an https page.

3.3.6.2 merchinvno:

The Merchant Transaction Reference is used as a reference key to the Wallety Payment Server database to obtain a copy of lost/missing receipts. And to reference the merchant's system with our system in Wallety Merchant Portal. It must be unique for each transaction attempt if it is to be used properly.

3.3.6.3 Gid

This is the merchant's gateway identifier, it tells the Wallety Payment Server which gateway to load and which secure data to calculate upon.

3.3.6.4 Amount

This parameter sets the value of the transactions, this value cannot be 0 or below, and is expressed in the lowest form of the gateway's set currency.

3.3.6.5 Redirect

This donates where the user is directed after completing his/her transaction, and is where Wallety should output the result of the transaction process via a GET request (in the URL using a query string). This has to be a fully qualified URL.

3.3.6.6 check_sum:

The checksum is an MD5 signature of the Wallety Secure Hash and the parameters in the Transaction Request. The inputs are concatenated as a single string starting with the Wallety Secure Hash, then each data field in ascending alphabetical order of that fields name, with no separators and no terminating character. This field is required and cannot be neglected neither replaced by the Wallety Secure Hash.

3.3.6.7 Additional optional:

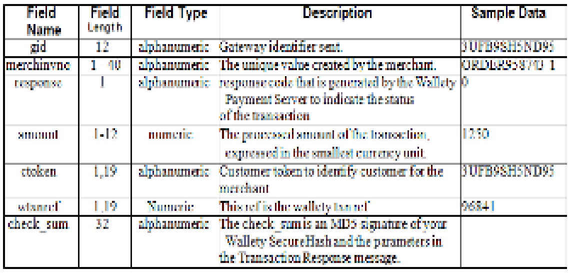
fields that are currently existing, however usage and attributes are not final are the Ticket Number field and the Expiry Fields, where the Ticket Number should hold the ticket number of a sold ticket (for example airline ticket numbers) and the expiry field which denotes the number of seconds this order should be valid for (our current values are from 300 to 3000).

And additional field for a generated value by the merchant to add security and prevent hacking using the Cross-Site Request Forgery (CSRF) will also be available.

4 Result Set

Currently Wallety only support response via a GET request (in the URL using a query string).

The following table 2 denotes the fields in the response.



Field Name	Field Length	Field Type	Description	Sample Data
gid	12	alphanumeric	Gateway identifier sent.	3UFB9SH5ND95
merchinvno	1 - 40	alphanumeric	The unique value created by the merchant.	ORDER958743-1
response	1	alphanumeric	response code that is generated by the Wallety Payment Server to indicate the status of the transaction	0
amount	1-12	numeric	The processed amount of the transaction, expressed in the smallest currency unit.	1250
ctoken	1,19	alphanumeric	Customer token to identify customer for the merchant	3UFB9SH5ND95
wtxnref	1,19	Numeric	This ref is the wallety txn ref	96841
check_sum	32	alphanumeric	The check_sum is an MD5 signature of your Wallety SecureHash and the parameters in the Transaction Response message.	

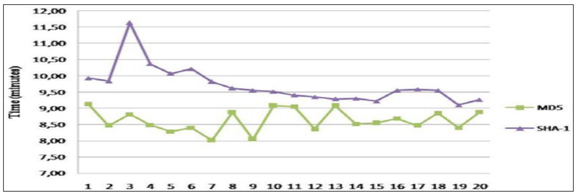


Fig 7 displays the average and the difference in estimation time to brute force. It can be seen that in the testing for 6 characters, the time difference is not too long, though the results still indicate that a brute force for SHA-1 takes longer. When performing a brute force password with a length of 7, 8 and 9 characters, the brute force time is getting longer. The difference in estimation time becomes the parameter that shows the complexity and strength of SHA-1 algorithms which is complex when compared with MD5.

4.1 wtxnref:

This field is very important and we highly recommend you to save it for further reference, although it will always be available to you in your portal.

This is the field that unites all entities to point to the same transaction. It is generated by Wallety, and is the key reference to the transaction in Wallety for the customer, merchant and bank. It is also accessible beyond Wallety's domain through the vpc_MerchTxnRef field on the Virtual Payment Client portal.

Wallety's Response should not be predicted, neither confirming payment transactions before Wallety responds with a successful code in the "response" field.

4.2 Report and Evaluation

The report request allows you to search for a previous transaction receipt. The search is performed on the key merchinvno, so the merchinvno field must contain a unique value.

If a transaction receipt is found, the results will contain the same fields as the original receipt plus the two flags described below.

report always returns these two flags in addition to the base Result Set mentioned earlier in the document:

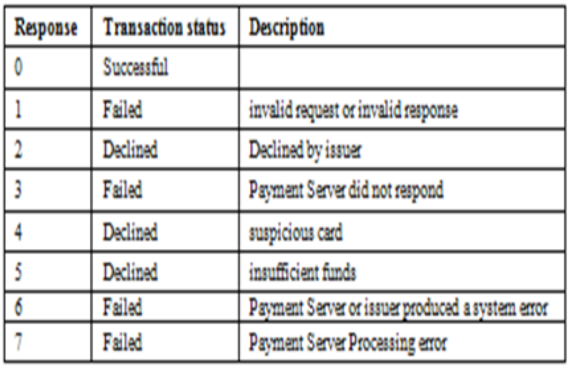
txn_exists: if no transactions are found that match the merchinvno number, this value will be set to "N" for No. If any transactions are found that match the merchinvno number, this value will be set to "Y" for Yes.

txn_duplicate: This is used to determine if there are multiple results. If the value is "N", then only one merchinvno matches the search criteria. If the value is "Y", then there are multiple merchinvno matching the search criteria, but it will return the most recent transaction. If the query result returned is not the correct one, the merchant must manually search through Merchant Administration.

5 Conclusion & Future Work

This paper investigated using different Encrypted Algorithms The returned response code from wallety.com after a transaction takes place, notifies the status of the transaction in addition, it could describe the reason of failed or declined transactions.

Any response other than 0 is a declined or failed transaction the response codes generated by the Payment Server are:



Response	Transaction status	Description
0	Successful	
1	Failed	invalid request or invalid response
2	Declined	Declined by issuer
3	Failed	Payment Server did not respond
4	Declined	suspicious card
5	Declined	insufficient funds
6	Failed	Payment Server or issuer produced a system error
7	Failed	Payment Server Processing error

Most of payment systems described above offer a secure means directly related to transfer credit/debit details for settlement in the existing financial systems. This also suffers from transaction processing costs, ensuring that low value transactions cannot be cost-effective. Well known institutions are able to aid in EPS (electronic payment system) adoption through the provision of a large installed base of customers. This study has also found that these institutions play other crucial roles in EPS adoption. Large partners are able to provide EPS with association with trusted brand names and marketing boom. These result in the system gaining credibility and public awareness. Once this has been achieved the system is assessed by users on factors such as simplicity, security and mutuality of stakeholder benefits.

E-commerce on the Internet needs payment mechanisms that can serve for as much diversity as commerce in the real world. Large value transactions will require secure ways to use existing bank card mechanisms.

References

1) Zarouk, Y. (2013). Consumer Protection of Civil Risk of Electronic Contracting - Comparative Study. Journal of Politics and Law, No. 9, p. 134.

2) Al-Sarayra M. (2009). The Legal Framework of the Contract through Electronic Means of Communication Study in Jordanian Legislation, Damascus University. Journal of Economic and Legal Sciences, Vol 25, No. 2, 2009, p. 825.

3) Bhatnager, A., Misra,S. and Rao, H.R. (2000). "On risk, convenience, and internet shopping behaviour". Communications of the ACM, vol. 43, pp 98-106

4) Caldwell, K. (2000). "Global electronic commerce-moving forward," Commerce Net: The Public Policy Report, vol. 2, pp. 2-17.

5) Farrell, Y.L., and G. Farrell, (2000). "A study on consumer fears and trust in internet based electronic commerce," in Proceedings of 13th International Bled Electronic Commerce Conference, pp. 647-658.

6) Friedman, M., Kahn, P.H. and Howe, D.C. (2000). "Trust online," Communications of the ACM, vol. 43, pp. 34-40.

7) Giff, S. (2000). "The influence of metaphor, smart cards, and interface dialogue on trust in ecommerce," M.Sc Project, University College London vol. 1, pp. 27-45.

8) Adams C. and Lloyd, S. (1999). Understanding Public-Key Infrastructure: concepts, standards, and deployment considerations, New Riders, Indianapolis: Macmillan vol. 23, pp. 13-30.

9) United Nations (2015). Resolution adopted by the General Assembly on 22 December 2015. Seventieth session Agenda item 18 (a) vol. 3, pp. 92.

10) OECD, (2016). Recommendation of the Council on Consumer Protection in E- commerce (Paris), available at http://www.oecd-ilibrary.org/industry-and- services/oecd-recommendation-of-the-council-on-consumer-protection-in-e-commerce_9789264255258-en (accessed 18 April 2017).

11) Neacsu, N.A. (2016). Consumer protection in electronic commerce, United Nations Conference on Trade and Development, Vol. 9, No. 1, pp. 301-308.