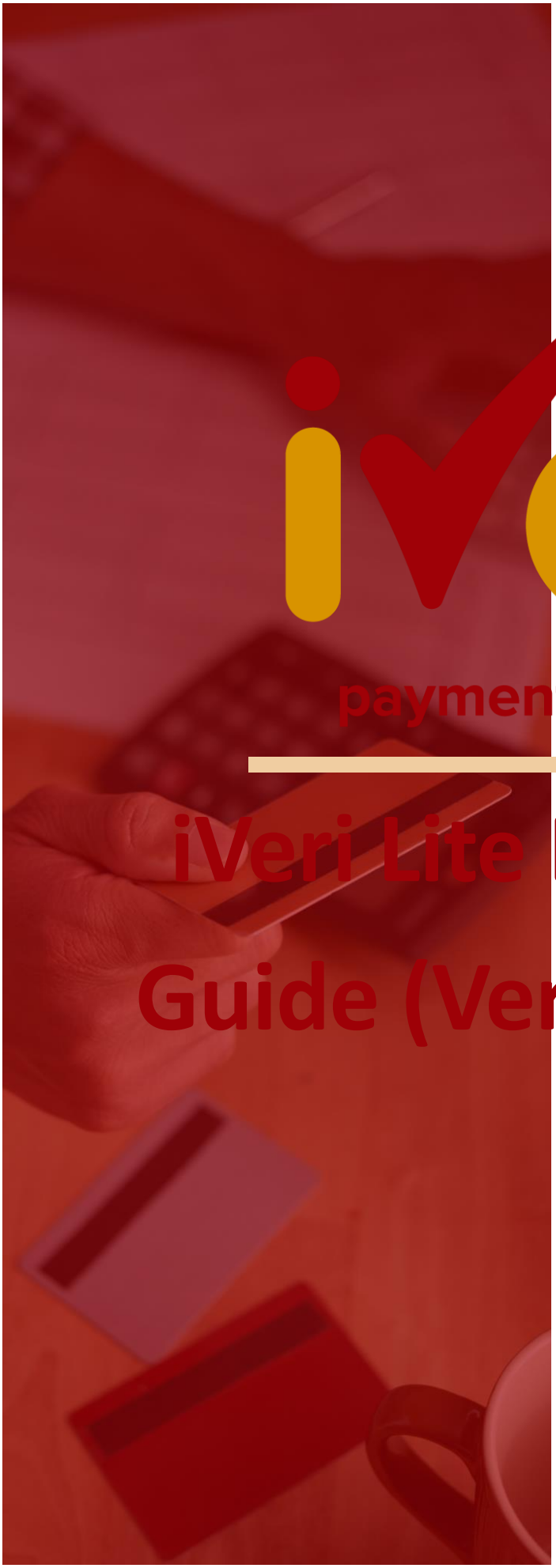
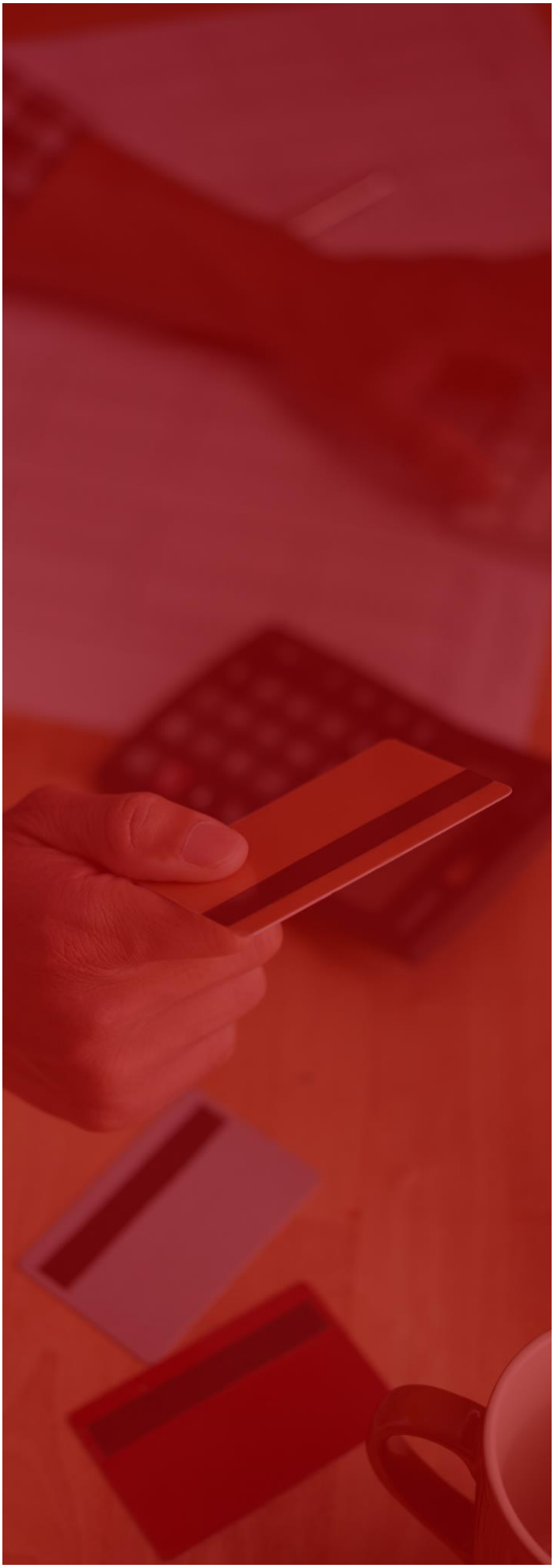
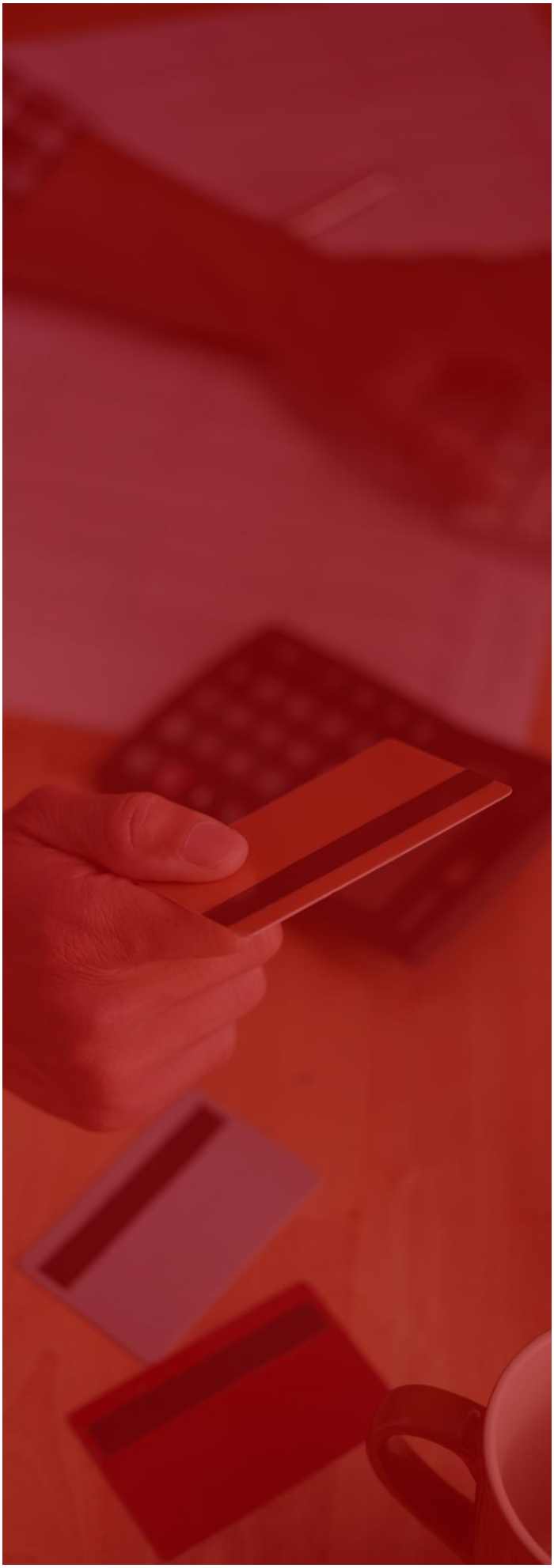


payment technology

iVeri Lite Developers Guide (Version 4.0.13)



Contents



REVISION HISTORY

Version	Author	Date	Description
3.0.0	Louise Gilbert	August 2008	Initial document (rewritten from version 2 of the iVeri lite documentation)
3.0.1	Louise Gilbert	September 2008	Updated after feedback from Eugene, Roland, Florence and Jeff
3.0.2	Louise Gilbert	September 2008	Updated after feedback from Roland
3.0.3	Roland Elferink, Louise Gilbert	October 2008	Updated after the development of the example's website and the addition of 3D secure to this document
3.0.4	Louise Gilbert	October 2008	Updated after feedback from Roland in respect of 3D Secure
4.0.0	Louise Gilbert	October 2008	First release
4.0.1	Eugene Kriek	July 2011	Added Mobile Money / Tokenized transaction / Lite+
4.0.2	Eugene Kriek	July 2012	Removed Lite and renamed Lite+ to Lite
4.0.2	Caryn Bessarabia	July 2012	Changed all diagrams to Lite+
4.0.3	Eugene Kriek	February 2013	Added a Webpage to get Transactional Information. Added recurring indicator: **Lite_Recurring_Payment**
4.0.4	Eugene Kriek	September 2014	Added Passenger Information to Line Items Added Airline Data
4.0.5	Eugene Kriek	June 2015	Updated the document with the extra parameters returned by a Authorisation Information request.
4.0.6	B. Habe	June 2016	Added Visa Check Out with iVeri Lite, Process & Flow and an Illustration
4.0.7	B.Habe	August 2016	Added Lite_Transaction_Token parameter; Added Generating the Token, Logic, Prerequisites, and Example code
4.0.8	B.Habe	September 2016	Changed Hosted and Nedbank Gateway URL's
4.0.9	B.Habe	January 2017	Added CSC Bank & CSC24Seven as a distributor including Gateway address for CSC
4.0.10	B Habe	April 2017	Updated the document to include the use of TransactionIndex on Subsequent Transactions
4.0.11	B. Habe	August 2017	• Renamed **Generate Token** to **"Merchant Token Generation"** • Added an Overview of the iVeri Lite – Hosted payment Page and various integration options available • Added details about the Litebox Hosted payment page Integration
4.0.12	B Habe	June 2019	• Added CIM Gateway URL's • Removed nPay related topics • Renamed Merchant Token Generation to "Merchant Hash Token Generation" • Renumbered all the sections • Updated Technical email address from support@iveri.co.za to operations@iveri.com
4.0.13	B Habe	October 2020	Updated the Lite version from 2.0 to 4.0

1. AIM

This document aims to show what is needed to complete the integration of a website with iVeri Lite hosted payment page to accept card transactions. The operation of 3D secure (Secure Code or Verified by Visa) as well as the interaction with iVeri Lite is described in this document, including the configuration required to implement 3D secure. This document is intended as a guide to the logic involved in making online transactions.

2.1 Overview of IVeri Lite - The Hosted Payment Page

The iVeri Lite hosted payment page is an ecommerce solution that is geared for merchants that want card acceptance in their online stores. The acceptance of various card schemes and different payment methods available is dependent on the acquiring bank and or merchant agreements.

The iVeri Lite hosted payment page can be integrated in 1 of 3 ways:

1. ***Full Redirect*** - A full Redirect to the Hosted payment page shifts the interaction of the shopper/buyer away from the merchant's website and only goes back to the merchant website when the transaction is complete.
2. ***LiteBox***- The LiteBox hosted payment appears or pops up within the merchant's website, the merchant's website remains unchanged, providing for a more user friendly and seamless checkout experience for both cardholder and merchant.
3. ***Shopping Carts*** - The iVeri Lite hosted payment page is integrated to some of the commonly used shopping carts. Merchants can download the iVeri Lite payment gateway extensions by contacting assist@iveri.com

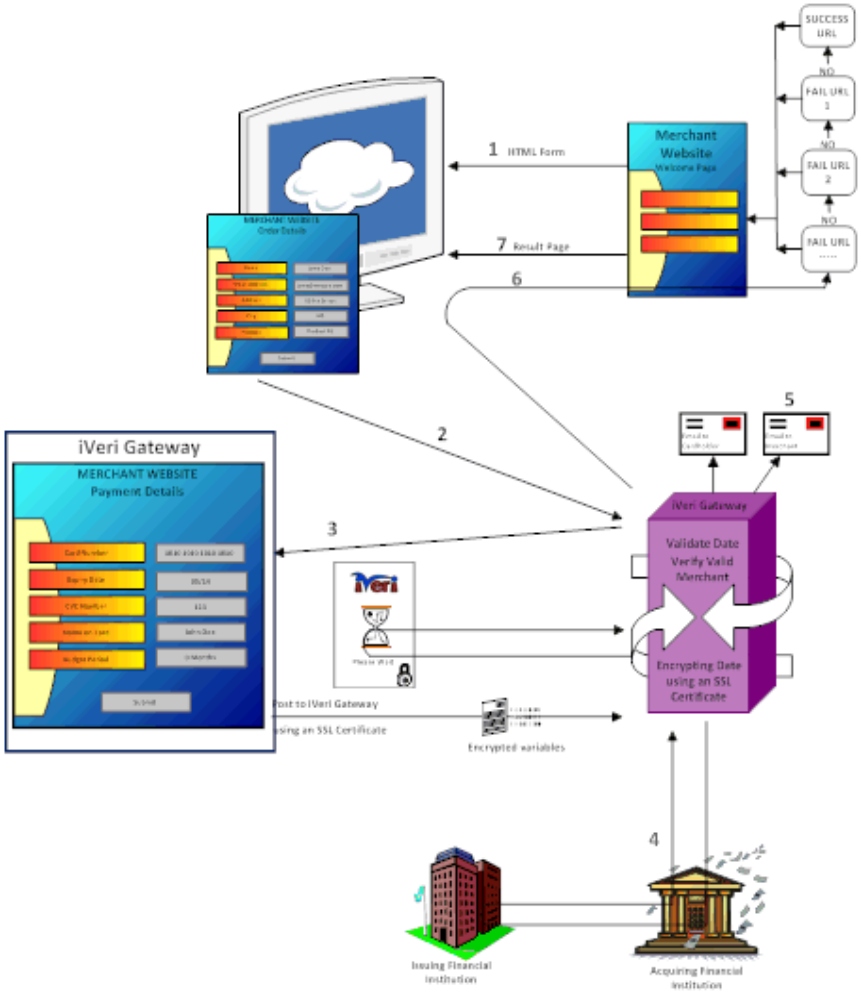
2. INTRODUCTION TO THE IVERI LITE

iVeri Lite requires very little integration and is aimed at Internet merchants who have limited technical resources. Lite transactions are processed on a web site and secured via an SSL certificate without the merchant having to buy SSL since iVeri lite takes care of it on their behalf. Although ideal for websites with small catalogs, Lite still provides a powerful processing engine.

2.1 iVeri Lite Process flow

The diagram below illustrates the flow of events of an iVeri Lite transaction:

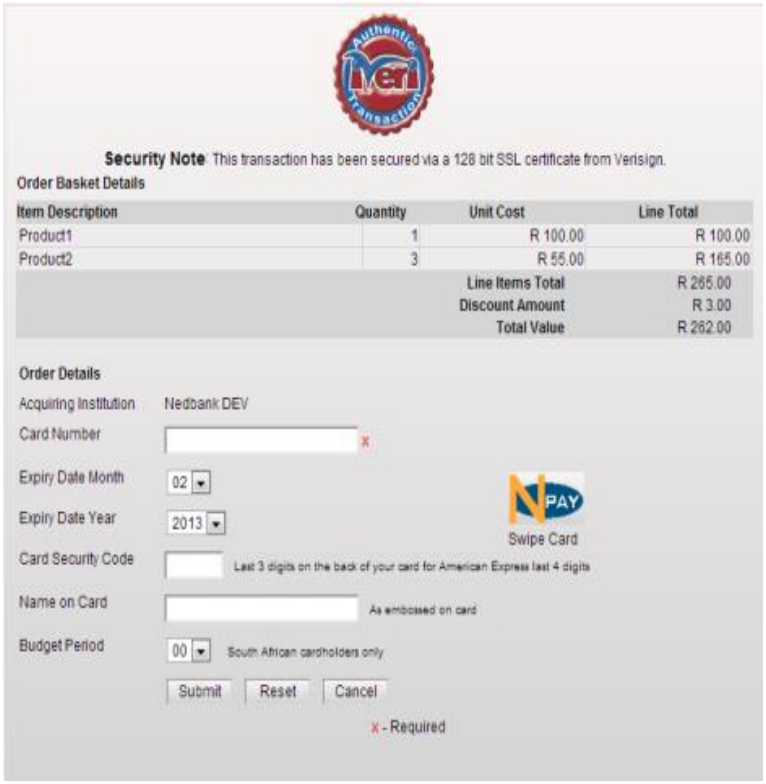
Illustration 1



1. The Card holder's browser is sent an HTML form via the Merchant's website, where he is prompted to enter transaction details like the billing and shipping details. The card holder is at the point in the purchase process where the basket has already been selected and he is now on the brink of actually paying for it. The web site thus knows the price of the basket, the Invoice Number (AUTOGENERATE, the merchant could also have iVeri BackOffice allocate the Invoice number for them) .

Confirm Details		
Name:	Mr. John Doe	
Email:	jdoe@mail.com	
Ship To Street (Line 1):	50 Sunny Drive Avenue	
Ship To Street (Line 2):	Sunsetville	
Ship To City:	Johannesburg	
Ship To Province:	Gauteng	
Ship To Postal Code:	2185	
Merchant Reference:	AUTOGENERATE	
Line Items		
Product Description	**Quantity**	**Unit Cost**
Example Product#1	1	10000
Sample Product#2	3	5500
Discount Amount:	**200**	
Total Order Amount:	**26300**	
Submit		

2. Once the Submit button has been clicked, the customers browser and captured details are sent to the Gateway verifying that the applicationID is valid, validating the variables i.e. purchase amount is an integer and that all required fields have been populated.



.

3. The Customers card details are captured on a SSL secured website where after clicking on the **Submit button** validation will of the entered values i.e. credit card number, expiry date will take place. Once validation has succeeded a “Please wait .....” message is returned to the card holder while the transaction is in process

   Authorization is performed at the respective acquiring financial institution/department who will then interact with the issuing financial institution/department.
4. The return of an authorization can either be successful, failed, system error or please try again later. A redirect is sent to the card holders browser telling it which Result URL to display (Example result page code).
5. In the event of a successful/error transaction the card holder as well as the Merchant can be emailed a statement indicating the successful completion of the transaction (an example of which can be seen in iVeri Lite Examples). All transactions are stored in a database, so it is always possible to trace a transaction in the case of an email failure.

6. A resultant success/error page is returned to the card holder by the merchant to show the transaction type, transaction amount, authorization code, merchant reference and purchase date.

3. GETTING STARTED WITH IVERI LITE

3.1 Registering as a merchant

Merchant account can be attained by registering with an acquiring bank, a list of which can be found under APPENDIX A.

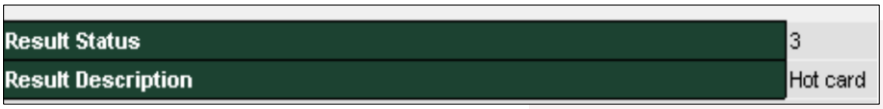
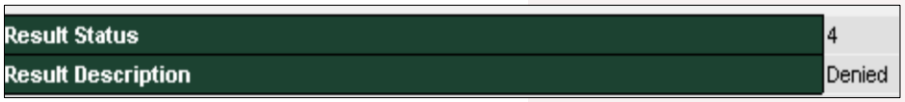
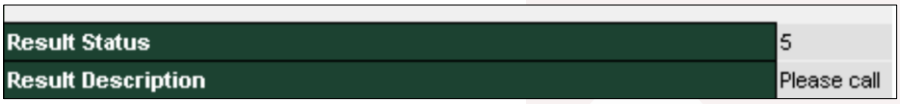
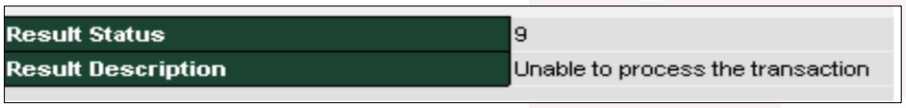
3.2 Development phase

For code examples refer to iVeri lite code examples, iVeri Lite result page code example and iVeri Lite Specification in APPENDIX B.

3.3 Test phase

The iVeri Gateway provides a mechanism where a merchant can perform test transactions that are routed to an iVeri Gateway issuer simulator. This enables a merchant to complete testing within the test environment. When the merchant is ready to Go LIVE, the acquiring bank can activate the merchant profile for LIVE processing which will be routed to the genuine card issuer.

When performing a test transaction, using your test application ID, the following credit card numbers must be used:

Credit card number	Expected result
4242424242424242	Returns “” (code 0)
2121212121212121	Randomly returns “Hot Card” (code 3)  “Denied” (code 4)  “Please call” (code 5) 
5454545454545454	Randomly returns “Unable to process” (code 9) times out 
All other card numbers	The emulator will return “Invalid card number” (code 14). “1111222233334444” for instance will produce this error. A real Credit Card Number will also generate this error code in Test.

NOTES:

The Expiry Date of the above test card numbers should be any date in the future.

Card Security Code (CVV) Field – you can use any three numbers for testing purposes.

If you want to generate the error “Invalid Expiry Date”, make the expiry date in the past.

3.4 Live phase

An email with a subject “Request to Go Live” with the following information needs to be sent to the distributor (contact details can be found under APPENDIX A. We need to be able to test your website's integration of iVeri Lite while it is still utilizing the Test ApplicationID.

- Merchant's Website URL
- User Group Number / Merchant number
- Test login credentials (Username / Password) for logging in to the merchant website as an end-user. NB: It is time consuming for our test team to register as users on each website that we have to test hence this is required by them. If no log-in required, please state so within your email.
- A detailed instruction on how to buy products from your website.

Once the distributor has tested your iVeri Lite integration, the distributor will enable the live ApplicationID. At this point, the merchant must replace the test ApplicationID used during testing with the live ApplicationID and deploy the code into the merchant's production environment.

4. VISA CHECK-OUT WITH IVERI LITE

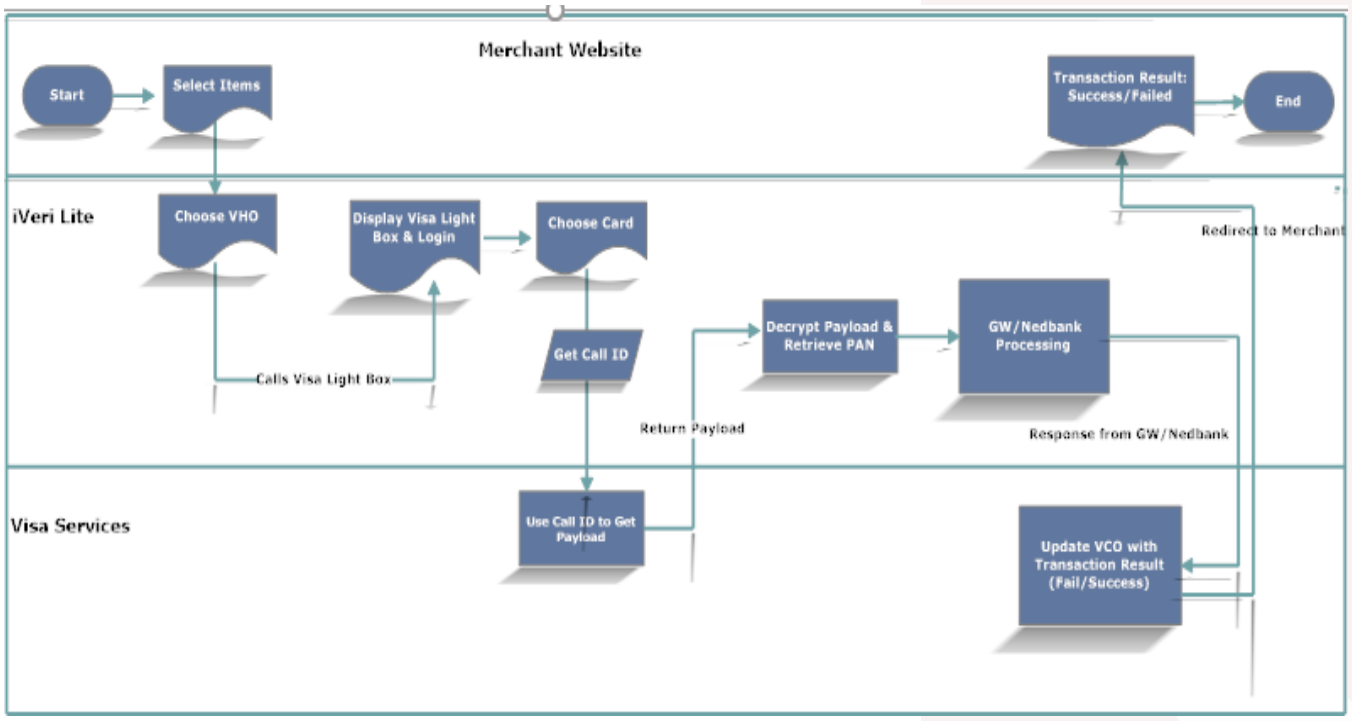
Visa Check-Out is a digital representation of a cardholders Visa Card. Cardholders can register their debit or credit cards by downloading the Visa Check-Out app. Once cardholders have their profiles and card details loaded in Visa Check-out, they are able to make purchases at merchants who are accepting with Visa Check-Out payments.

The Visa Check Out process flow explained below showcases the different parties involved.

4.1 Process Flow in Visa Check-Out with iVeri Lite

1. Cardholder selects Visa Check-Out as payment method
2. iVeri Lite calls Visa Light Box or Widget and presents it to the Cardholder to Login
3. Cardholder Logins via Visa Light Box
4. Cardholder selects a card and presses **Continue**
5. Upon clicking Continue, iVeri Lite receives a Call ID (uniquely identifies the transaction)
6. iVeri Lite uses the Call ID to get the Payload from Visa services
7. Payload is returned by Visa services to iVeri Lite
8. iVeri Lite decrypts the Payload and retrieves the PAN
9. iVeri Lite sends the transaction request to the gateway/acquire
10. When the response is received, Visa services is updated with either a Failed or Successful response
11. Lastly, a redirect to the merchant with the transaction response is made.

4.2 Illustration of Visa Check-Out Process With iVeri Lite

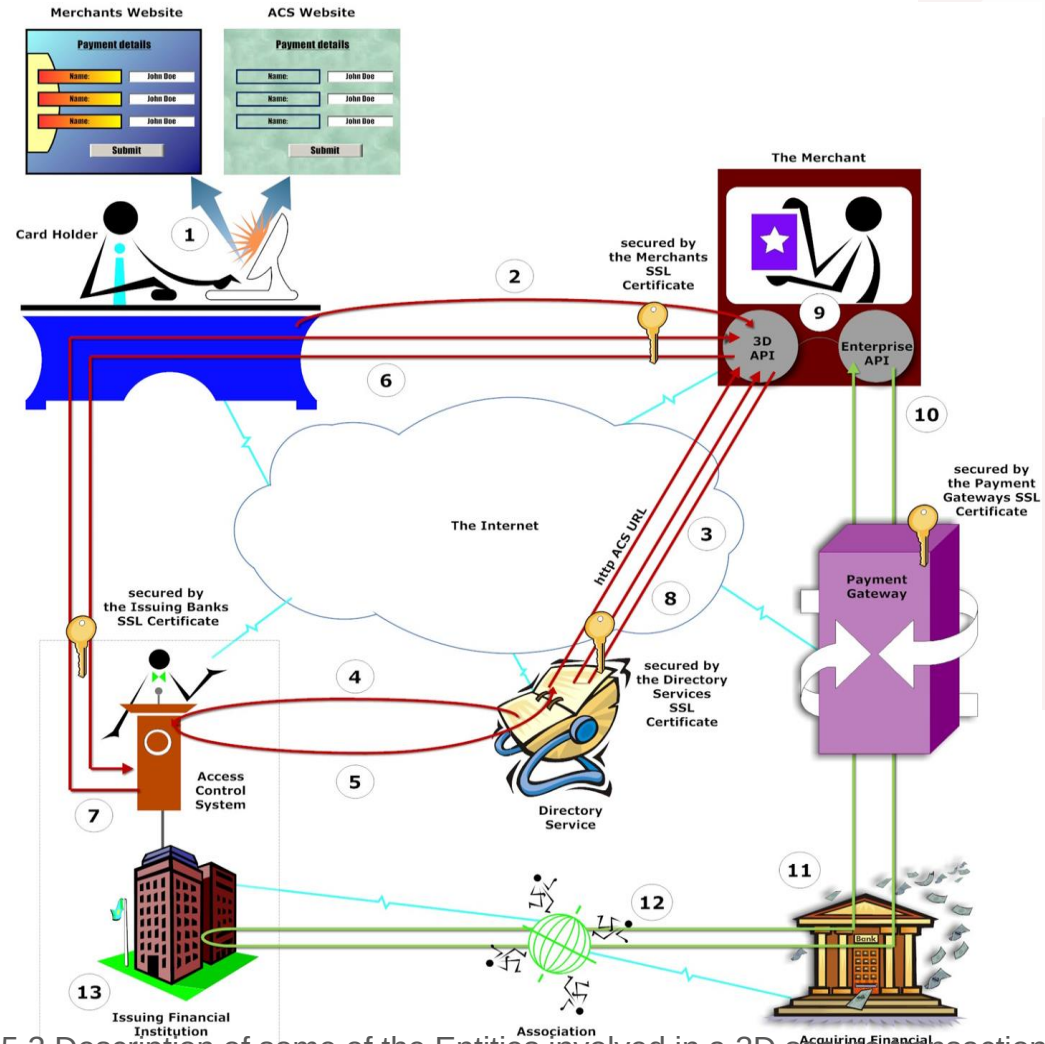


5. INTRODUCTION TO 3D SECURE

3D secure (Secure Code or Verified by Visa), is all about call back protection and making sure card holders are who they say they are. The Illustration and process that follows, shows 3D secure in use as implemented by Merchants using iVeri Enterprise and not iVeri Lite. Illustrating the process in this way, helps to show how each entity is involved even if certain functions are performed on behalf of Merchants using iVeri Lite.

5.1 Process flow of a 3D secure transaction

Illustration 3



5.2 Description of some of the Entities involved in a 3D secured transaction

An **Association**, i.e. Visa or Master Card resides between the Acquiring and Issuing financial institutions (Banks). The Association has links and agreements between the respective banks, so by virtue of the Issuer having an agreement with the Association they in effect have an agreement with the Acquire who also has an agreement with the Association.

In addition to the Associations, South Africa has an entity called **BANKSERV** which processes local card transactions on behalf of certain South African banks.

All local transactions are sent through BANKSERV but as far as the Merchant is concerned, his interaction with BANKSERV is transparent.

BANKSERV also hosts the **Access Control System** and **Directory Service** (discussed in more detail later) on behalf of curtain banks in South Africa.

BANKSERV operates in the space of cooperation between the banks in terms of clearing and settlement and is non-competitive to its shareholders and customers.

Merchants need to register their Merchant ID with BANKSERV.

The Internet is used to network most of the entities needed for a 3D secure transaction.

When a Merchant integrates 3D secure into their Website, they need to integrate a **3D secure API** as well as an **Enterprise API**.

5.3 3D secure transaction process flow

1. When the **Card Holder** is at the point where he has chosen a basket of goods and is on the brink of paying for it, he is presented with, amongst others a Credit Card Number field, Expiry Date field and CVV field which he fills in and
2. after clicking apply the Card Holders details are sent to **the Merchants** 3D API thereby starting a conversation between the Merchant and Card Holder.
   The Merchant wants to protect himself against charge backs so he will perform a 3D secure transaction as opposed to a normal transaction.

If the Merchant did not want the protection, he would simply send the form variables/tag names straight to Enterprise API and from there it would (following the green flow lines) go to the Payment Gateway – Acquire – Association – Issuer and back with an approved response but without any charge back protection.

3. To Continue through with an enterprise 3D transaction (following the red flow lines), the Merchant now has amongst others the Card Holders Credit Card Number, Expiry Date and CVV number. He uses the 3D API to send a message to the Directory service (who governs the charge back domain) asking two questions. Firstly, is the Issuer of the card enrolled for 3D secure? If NO, the API gets an immediate response.
4. If yes, the **Directory Service** will ask the **Access Control System (ACS)** of the Issuing Bank if the specific card is enrolled for 3D secure?
5. If YES, the Directory Service will respond to the Merchant with the URL of the issuing banks ACS.

   if NO, it will respond to the Merchants 3D API with, “the Card Holder is not enrolled but he should to be”.

   The Merchant now needs to make two decisions, does he stop the transaction since he now knows the card is completely uncovered or does, he takes a chance?
6. Let's say he doesn't want to take any chances; he uses the ACS URL to redirect the Card Holders browser from the Merchants website to the actual ACS website of the Issuing Bank. From the Card Holders perspective, he has gone from being on the merchant’s website to being on his banks ACS website. There are 2 situations that may occur here. Firstly, the Card Holder is enrolled, in which case he will be asked if he would like to continue with the transaction, if so, to enter his password and press submit. If the Card Holder is not enrolled, he will be asked to fill in specific details to which his bank will respond with a sms to validate that the Card Holder is who he says he is. The Card Holder will then use the password received in the sms and press submit if he elects to continue with the transaction.
7. Clicking submit will do another redirect back to the Merchants website with 1 of a few responses. 1 The Card Holder canceled so stop the transaction. 2 If the Card Holder has entered his password and been validated by the issuing bank, the Merchant will get a certificate and other pertinent information in the response.

8. The Merchant then sends the information to the directory service again, requesting it to validate that the information is not fraudulent. This normally checks out fine, but it needs to be done none the less.
9. The form variables/tag names and information added when the ACS issued the validation certificate are sent to the Enterprise API and (following the green flow lines) onto the **10**: The Payment Gateway – **11**: the Acquirer – **12**: the Association – **13**: the Issuer – and back.

Since every entity has validated that the Card Holders details are correct and that funds are available, the risk of charge back is greatly reduced. Effectively where the risk of a card not present scenario would lie with the Merchant it now lies with the Card Holder, this is due to the Merchant taking the trouble to integrate 3D secure to verify the Card Holders details through a certificate received from the Issuer that has been validated by the Directory Service and the Merchant is assured that the Card Holder has funds (by the green flow lines). This way the Merchant is almost 100 % assured against card not present fraud.

6. APPENDIX A

6.1 Distributors

Below is the contact information per distributor for registering billing and banking details with iVeri.

An iVeri Distributor markets the services of the iVeri Gateway and products within a locality.

6.1.1 Nedbank South Africa

Location	South Africa	
Telephone	0860 114 966	
Web sites	http://www.iveri.co.za	http://www.nedbank.co.za
Email	Technical Assistance	operations@iveri.com
	Non-technical requests/questions (e.g. costs, agreements, product information etc)	info@iveri.co.za

6.1.2 Nedbank Namibia

Location	Namibia	
Web sites	http://www.iveri.co.za	http://www.nedbank.co.za
Email	Technical Assistance	operations@iveri.com
	Non-technical requests/questions (e.g. costs, agreements, product information etc)	info@iveri.com

6.1.3 I&M Bank

Location	Kenya	
Web sites	http://www.iveri.com	http://www.imbank.com/KE/
Email	Technical Assistance	operations@iveri.com
	Non-technical i.e. product information	info@.iveri.com

6.1.4 CBZ Bank

Location	Zimbabwe	
Web sites	http://www.iveri.com	http://www.cbzbank.co.zw/
Email	Technical Assistance	operations@iveri.com
	Non-technical i.e. product information	info@veri.com

6.1.5 CSC & CSC24Seven Bank

Location	Lebanon & Cyprus	
Web sites	http://www.iveri.com	http://www.cscgroup.com
Email	Technical Assistance	operations@iveri.com
	Non-Technical i.e. product information	info@.iveri.com

6.1.6 CIM

Location	Lebanon & Cyprus	
Web sites	http://www.iveri.com	http://www.cim.mu
Email	Technical Assistance	operations@iveri.com
	Non-technical i.e. Product information	info@.iveri.com

6.2 Gateway addresses

6.2.1 Nedbank Gateway

Website	URL	Port
BackOffice	https://backoffice.nedsecure.co.za	443
Lite	https://portal.nedsecure.co.za/Lite/Authorise.aspx	
Authorisation Information	https://portal.nedsecure.co.za/Lite/AuthoriseInfo.aspx	

6.2.2 Hosted Gateway

Website	URL	Port
BackOffice	https://backoffice.host.iveri.com	443
Lite	https://portal.host.iveri.com/Lite/Authorise.aspx	
Authorisation Information	https://portal.host.iveri.com/Lite/AuthoriseInfo.aspx	

6.2.3 CSC Hosted Gateway

Website	URL	*Port*
Backoffice	https://backoffice.cscacquiring.com	443
Lite	https://portal.cscacquiring.com/Lite/Authorise.aspx	
Authorisation Information	https://portal.cscacquiring.com/Lite/AuthoriseInfo.aspx	

6.2.4 CIM Hosted Gateway

Website	URL	*Port*
Backoffice	https://Backoffice.merchant.cim.mu	443
Lite	https:// portal.merchant.cim.mu/Lite/Authorise.aspx	
Authorisation Information	https:// portal.merchant.cim.mu/Lite/AuthoriseInfo.aspx	

7. APPENDIX B

7.1 iVeri Lite Email examples

Email to Merchant

This is an order confirmation from the <DISTRIBUTOR> Payment Gateway to <MERCHANT NAME> Lite Application.

We have received the following Sale from your customer:

Customer Details:

Name : Mr. John Doe

Email : John

Payment Details

Transaction Type : Sale

Transaction Index : A0394EB4-BBC1-4567-BBCB-A56B702050DD

Merchant Reference : LITE0000028

Card Number : 4242.....4242

Expiry Date : 092009

Acquirer Reference : 80903:09089990

Electronic Commerce Indicator : SecureChannel

Order Details

Purchase DateTime : 2008-09-03 09:08:32

Total Order Amount : R 100.00

Line Item Details:

Item Description	Quantity	Unit Cost	Line Total
Donation product#1	1	100.00	100.00

Additional Info:

Please contact <DISTRIBUTOR> should you have any queries.

Email to card holder

Dear Mr. John Doe

This is to confirm that <MERCHANT NAME> Lite Application has made a Sale on your card (number: 4242.....4242) for R 0.99.

The transaction ocurred at 2008-08-14 10:13:27 (South Africa Standard Time).

The reference number used by <MERCHANT NAME> Lite Application was <App id>

Order Details

Purchase DateTime : 2008-08-14 10:13:27

Total Order Amount : R 0.99

Line Item Details:

Item Description	Quantity	Unit Cost	Line Total
Donation product#1	1	0.35	0.35
Donation product#2	2	0.32	0.64

Additional Info:

Expiry Date Month : 07

Expiry Date Year : 2011

Should you have any concerns about this transaction or the services offered by <MERCHANT NAME>'s Lite Application, please contact <MERCHANT NAME>'s Lite Application directly.

7.2 iVeri Lite Specification

The following parameters are expected in the form submitted to the Gateway at step 3 of the iVeri Lite process flow

and/or returned by the Gateway in the response at step 6 to the cardholder via the merchant's website.

Note: Not all of the fields in this specification are mandatory. Fields which are marked in grey are the only mandatory fields.

General descriptor	Name	length	values	In step 3 or step 6 of illustration 1	Notes
Version	Lite_Version	5	4.0	3 and 6	Set to current Version number ex. 2.0, for Old Version left blank – Not available.
iVeri Status	Lite_Payment_Card_Status	10		3 and 6	Status of transaction received in response
iVeri Application id	Lite_Merchant_ApplicationId	36		3 and 6	Allocated to the Merchant during registration
Amount to authorize	Lite_Order_Amount	10		3 and 6	The total amount for the order including tax in cents. This must be equal to the sum of the lineamount*linequantity for each line item
MerchantTrace	Lite_Merchant_Trace	64		3 and 6	Optional. Unique merchant identification for the request. This value can be used by the merchant to confirm the status of the transaction if need be. Appendix D
CardHolderPresence	Lite_Recurring_Payment	5	False	3 and 6	Set to True or False. If True, then If your application id has been enabled for Recurring transactions then this will be the first transaction. **Note:** Your **application id** has to be enabled for recurring payments otherwise this indicator will be ignored.
Terminal on the Web	Lite_Order_Terminal	8		3 and 6	Optional
Discount	Lite_Order_DiscountAmount	10		3 and 6	this is the discount field for iVeri lite, this field should be used as a discount field for the entire shopping basket. Please make use of this field and remember to adjust the Lite_Order_Amount
Authorisation Code	Lite_Order_AuthorisationCode	6		3 and 6	Optional preauthorisation code received from a financial institution
Budget Period	Lite_Order_BudgetPeriod	2	0	3 and 6	Optional request to put this order onto a budget plan. The normal options for this are 0, 6, 12, 18, 24 and 36. 0 indicating to budget period. Specifying this field does not guarantee that the request will be granted a budget period.
Success End Url	Lite_Website_Successful_Url	255		3 and 6	URL to pass control back to upon successful completion of a transaction. Control of the transaction is passed back to this URL if the Lite_Payment_Card_Status = 0
Failed End Url	Lite_Website_Fail_Url	255		3 and 6	URL to pass control back to if the authorisation is refused by the financial institution. Control of the transaction is passed back to this URL if the Lite_Payment_Card_Status is anything other than 0, 1, 2, 5, 9 or 255
Network error End Url	Lite_Website_TryLater_Url	255		3 and 6	URL to pass control back to if there is a system error which could be overcome by trying again later. Control of the transaction is passed back to this URL if the Lite_Payment_Card_Status = 1, 2, 5 or 9
Code error End Url	Lite_Website_Error_Url	255		3 and 6	URL to pass control back to if the form has not been filled in correctly of there is an inconsistency in the form. Control of the transaction is passed back to this URL if the Lite_Payment_Card_Status = 255
LineItem	Lite_Order_LineItems_Product_#	255		3 and 6	Line item of what is being ordered. The # indicates a number starting at 1 and incrementing by 1 for every line item. This is essentially a description of the item being ordered.
LineQuantity	Lite_Order_LineItems_Quantity_#	10		3 and 6	Line item Quantity of what is being ordered. The # indicates a number starting at 1 and incrementing by 1 for every line item. This is the number of units being ordered.
LineAmount	Lite_Order_LineItems_Amount_#	10		3 and 6	Line item cost including tax if any of what is being ordered. The # indicates a number starting at 1 and incrementing by 1 for every line item. This is the unit price for this particular item.
Autogenerate Invoice Extension	Lite_ConsumerOrderID_PreFix	5		3 and 6	If the Merchant sets the ECOM_CounsuremOrderID to "AUTOGENERATE" then this field is used to control the first 3 to 6 characters of the autogenerated consumerorderid E.g. "INV" set in this field will result in an autogenerated consumerorderid of "INV0001", "INV0002" and so on.
Card Pre-Auth Mode	Lite_Authorisation	5	False	3 and 6	Set to True or False. If True then a Authorisation will be made, if false a Debit will be made. The Authorisation code will be stored for 3 weeks before you need to confirm the transaction in the BackOffice under view orders.
Transaction Token	Lite_Transaction_Token	32			Merchant should generate the token: Encoded data should be Lite_Order_Amount, Lite_Merchant_ApplicationId, Ecom_BillTo_Online_Email & TimeStamp
ship to title	Ecom_ShipTo_Postal_	4		3 and 6	e.g., Mr., Mrs., Ms.; field commonly not used

General descriptor	Name	length	values	In step 3 or step 6 of illustration 1	Notes
	Name_Prefix				
ship to first name	Ecom_ShipTo_Postal_Name_First	15		3 and 6	
ship to middle name	Ecom_ShipTo_Postal_Name_Middle	15		3 and 6	may also be used for middle initial
ship to last name	Ecom_ShipTo_Postal_Name_Last	15		3 and 6	
ship to name suffix	Ecom_ShipTo_Postal_Name_Suffix	4		3 and 6	e.g., Ph.D., Junior, Esquire; field commonly not used
ship to street1	Ecom_ShipTo_Postal_Street_Line1	20		3 and 6	
ship to street2	Ecom_ShipTo_Postal_Street_Line2	20		3 and 6	
ship to street3	Ecom_ShipTo_Postal_Street_Line3	20		3 and 6	
ship to city	Ecom_ShipTo_Postal_City	22		3 and 6	
ship to state or province	Ecom_ShipTo_Postal_StateProv	2		3 and 6	2 characters are the minimum for the US and Canada, other countries may require longer fields; for the US use 2-character US Postal state abbr
ship to zip or postal code	Ecom_ShipTo_Postal_PostalCode	14		3 and 6	
ship to country	Ecom_ShipTo_Postal_CountryCode	2		3 and 6	use ISO 3166 2 letter codes for country names
ship to phone	Ecom_ShipTo_Telecom_Phone_Number	10		3 and 6	10 digits are the minimum for the US and Canada, other countries may require longer fields, recommend placing on "x" before an extension
ship to email	Ecom_ShipTo_Online_Email	40		3 and 6	e.g., jsmith@isp.com
bill to title	Ecom_BillTo_Postal_Name_Prefix	4		3 and 6	e.g., Mr., Mrs., Ms.; field commonly not used
bill to first name	Ecom_BillTo_Postal_Name_First	15		3 and 6	If this is blank your invoice would read Dear Customer.
bill to middle name	Ecom_BillTo_Postal_Name_Middle	15		3 and 6	may also be used for middle initial
bill to last name	Ecom_BillTo_Postal_Name_Last	15		3 and 6	
bill to name suffix	Ecom_BillTo_Postal_Name_Suffix	4		3 and 6	e.g., Ph.D., Junior, Esquire; field commonly not used
bill to street1	Ecom_BillTo_Postal_Street_Line1	20		3 and 6	
bill to street2	Ecom_BillTo_Postal_Street_Line2	20		3 and 6	
bill to street3	Ecom_BillTo_Postal_Street_Line3	20		3 and 6	
bill to city	Ecom_BillTo_Postal_City	22		3 and 6	
bill to state or province	Ecom_BillTo_Postal_StateProv	2		3 and 6	2 characters are the minimum for the US and Canada, other countries may require longer fields; for the US use 2-character US Postal state abbr
bill to zip or postal code	Ecom_BillTo_Postal_PostalCode	14		3 and 6	
bill to country	Ecom_BillTo_Postal_CountryCode	2		3 and 6	use ISO 3166 2 letter codes for country names
bill to phone	Ecom_BillTo_Telecom_Phone_Number	10		3 and 6	10 digits are the minimum for the US and Canada, other countries may require longer fields, recommend placing on "x" before an extension
bill to email	Ecom_BillTo_Online_Email	40		3 and 6	This is the email address the invoice would be mailed to. e.g., jsmith@isp.com
receipt to title	Ecom_ReceiptTo_Postal_Name_Prefix	4		3 and 6	e.g., Mr., Mrs., Ms.; field commonly not used
receipt to first name	Ecom_ReceiptTo_Postal_Name_First	15		3 and 6	

General descriptor	Name	length	values	In step 3 or step 6 of illustration 1	Notes
receipt to middle name	Ecom_ReceiptTo_Postal_Name_Middle	15		3 and 6	may also be used for middle initial
receipt to last name	Ecom_ReceiptTo_Postal_Name_Last	15		3 and 6	
receipt to name suffix	Ecom_ReceiptTo_Postal_Name_Suffix	4		3 and 6	e.g., Ph.D., Junior, Esquire; field commonly not used
receipt to street1	Ecom_ReceiptTo_Postal_Street_Line1	20		3 and 6	
receipt to street2	Ecom_ReceiptTo_Postal_Street_Line2	20		3 and 6	
receipt to street3	Ecom_ReceiptTo_Postal_Street_Line3	20		3 and 6	
receipt to city	Ecom_ReceiptTo_Postal_City	22		3 and 6	
receipt to state or province	Ecom_ReceiptTo_Postal_StateProv	2		3 and 6	2 characters are the minimum for the US and Canada, other countries may require longer fields; for the US use 2-character US Postal state abbr
receipt to zip or postal code	Ecom_ReceiptTo_Postal_PostalCode	14		3 and 6	
receipt to country	Ecom_ReceiptTo_Postal_CountryCode	2		3 and 6	use ISO 3166 2 letter codes for country names
receipt to phone	Ecom_ReceiptTo_Telecom_Phone_Number	10		3 and 6	10 digits are the minimum for the US and Canada, other countries may require longer fields, recommend placing on "x" before an extension
receipt to email	Ecom_ReceiptTo_Online_Email	40		3 and 6	e.g., jsmith@isp.com
card type	Ecom_Payment_Card_Type	4		3 and 6	use the first 4 letters of the association name: American Express=AMER; Diners Club=DINE; Discover=DISC; JCB=JCB; Mastercard=MAST; Visa=VISA; other association names may require more than 4 characters. You can tell card numbers apart from the first digit of a credit card number. 3 - American Express 4 - Visa 5 - MasterCard
payment protocols available	Ecom_Payment_Card_Protocols	20	IVERI	3 and 6	A space seperated list of protocols available for a specified card. Initial list of case insensitive tokens: "none", "set", or "setcert". "Set" indicates usable with SET Protocol (i.e., is in a SET wallet) without a SET certificate. "Setcert" indicates same but does have a SET certificate. "None" indicates standard is being used but wallet is not SET capable or the card has not been entered into the SET wallet. Usually a hidden field.
consumer generated order ID	Ecom_ConsumerOrderID	20		3 and 6	unique order ID generated by the consumer software. If at all possible, this should be filled out. If it is impossible iVeri will generate one for you but it should preferably be filled out by the merchant. For iVeri to create a unique ordered for you please set this to "AUTOGENERATE" and refer to the Lite_ConsumerID_PreFix which enables you to specify the first few letters of the invoice.
schema version number	Ecom_SchemaVersion	30	1.0	3 and 6	Identifies the ecom schema version number; format 999_99; digit (3) _digit (2); defined within a URL (e.g. http://www.w3c.org/ecom/1_0). Field should be included on every page with an ECML field on it and is usually a hidden field.
end transaction flag	Ecom_TransactionComplete	-	FALSE	3 and 6	A flag to indicate that this web page/aggregate is the final one for this transaction. Usually a hidden field.
Description lite transaction	Lite_Result_Description	255		6	Status of transaction described, if any error – error description.
Index	Lite_TransactionIndex	36		6	This is a globally unique number generated by iVeri and is iVeri unique number to identify this transaction.
consumer generated order ID	MerchantReference	20		6	unique order ID generated by the consumer software. If at all possible, this should be filled out. If it is impossible iVeri will generate one for you but it should preferably be filled out by the merchant. For

General descriptor	Name	length	values	In step 3 or step 6 of illustration 1	Notes
					Veri to create a unique ordered for you please set Lite_ConsumerID to "AUTOGENERATE" and refer to the Lite_ ConsumerOrderID PreFix which enables you to specify the first few letters of the invoice.
Bank Reference	Lite_BankReference	11		6	Cycle, Trace number eg. 12345,12345
Transaction Date	Lite_TransactionDate			6	The date the authorisation took place
Referrer	Lite_Referrer			6	The Website Referrer where the merchants Transaction started.
Currency	Lite_Currency_AlphaCode	3		3 and 6	The Currency in which the transaction will be processed
Note: The below fields are required when Implementing section. **"11.*Tokenisation: TransactionIndex on Subsequent Transactions*"**					
PanFormat	Lite_PanFormat	64			Methodology that specifies how to obtain the PAN details
TransactionIndex	Lite_TransactionIndex	38			Unique identifier allocated by iVeri for a series of related transactions. If PANFormat is 'TransactionIndex', TransactionIndex is used to locate a previous transaction for the PAN to be resolved.
CardNumber	**Ecom_Payment_Card_Number**	20			A dotted Pan number as returned in a previous transaction would have to be passed in this field

Note: The following fields will be used when submitting to CyberSource for Advanced Fraud Screening and will not be saved to the iVeri system.

General descriptor	Name	length	values	In step 3 or step 6 of illustration 1	Notes
Passenger FirstName	Lite_Order_LineItems_PassengerFirstName_#	60		3 and 6	Passenger's first name.
Passenger LastName	Lite_Order_LineItems_PassengerLastName_#	60		3 and 6	Passenger's last name.
Passenger ID	Lite_Order_LineItems_PassengerID_#	32		3 and 6	ID of the passenger to whom the ticket was issued. For example, you can use this field for the frequent flyer number
Passenger Status	Lite_Order_LineItems_PassengerStatus_#	32		3 and 6	Your company's passenger classification, such as with a frequent flyer program. In this case, you might use values such as standard, gold, or platinum.
Passenger Type	Lite_Order_LineItems_PassengerType_#	32		3 and 6	Passenger classification associated with the price of the ticket. You can use one of the following values: • ADT: Adult • CNN: Child • INF: Infant • YTH: Youth • STU: Student • SCR: Senior Citizen • MIL: Military

Note: The following Airline addendum data is additional transaction data which appear on a card holder's statement when buying a ticket from an airline merchant who include this data in a transaction request.

General descriptor	Name	length	values	In step 3 or step 6 of illustration 1	Notes
Passenger Name	Airline_PassengerName	20		3 and 6	Passenger name as printed on ticket.
Primary TicketNumber	Airline_PrimaryTicketNumber	15		3 and 6	The ticket numbers.
First Departure LocationCode	Airline_FirstDepartureLocationCode	3		3 and 6	Code for departure airport location, eg. JNB for Johannesburg
First Arrival LocationCode	Airline_FirstArrivalLocationCode	3		3 and 6	Code for destination airport location, eg. JNB for Johannesburg
PNR Number	Airline_PNRNumber	6		3 and 6	
Office IATA Number	Airline_OfficeIATANumber	8		3 and 6	The office IATA number
Order Number	Airline_OrderNumber	8		3 and 6	The order number
Place of Issue	Airline_PlaceOfIssue	20		3 and 6	The ticket office location
Departure Date	Airline_DepartureDate	8		3 and 6	Date of departure in yyyymmdd format
Departure Time	Airline_DepartureTime	15		3 and 6	Departure time of the first leg of the trip. Use one of the following formats: • HH:mm \"GMT\"zzz • HH = two-digit hour in 24-hour format • mm = two-digit minutes • zzz = time zone of the departing flight, for example: If the airline is based in city A, but the flight departs from city B, z is the time zone of city B at the time of departure. Important For travel information, use GMT instead of UTC or use the local time zone. Examples 19:55 GMT+02:00 19:55 GMT+0200 11:25 GMT-03:00 11:25 GMT-0300 Note When specifying an offset from GMT, the format must be exactly as specified in the example. Insert no spaces between the time zone and the offset.
Complete Route	Airline_CompleteRoute	25		3 and 6	Concatenation of individual travel legs in the format ORIG1-DEST1[:ORIG2-DEST2...:ORIGn-DESTn], f or example:CPT-JNB :JNB-:NBO. For airport codes
Journey Type	Airline_JourneyType	25		3 and 6	Type of travel, for example: one way or round trip.

Note: The following fields will be captured when the merchant's website transfers control to the Lite web page on the iVeri Gateway.

General descriptor	Name	length	values	In step 3 or step 6 of illustration 1	Notes
card number	Ecom_Payment_Card_Number	19		3 and 6	The card number embossed on the cardholder's card
. cardholder verification value	Ecom_Payment_Card_Verification	4		3 and 6	additional cardholder verification number such as American Express' CIV, MasterCard's CVC2, and Visa's CVV2 values
card expiration date month	Ecom_Payment_Card_ExpDate_Month	2	1-12	3 and 6	Jan - 1, Feb - 2, March - 3, etc.
card expiration date year	Ecom_Payment_Card_ExpDate_Year	4		3 and 6	Value in wallet cell is always 4 digits, e.g., 1999, 2000, 2001.
name on card	Ecom_Payment_Card_Name	30		3 and 6	name of cardholder

NOTE: Please remember to do the relevant configuration of the card capture page in BackOffice. Refer to the Lite BackOffice user guide.

8. APPENDIX C: ADDITIONAL VARIABLES

Note: These are form variables that are not part of the iVeri Lite specification but may be required by the merchant.

These form variables (if used) will be returned to the website together with all the iVeri Lite form variables.

For added security, to prevent possible fraud by someone obtaining the websites return URL and parameters, **we highly recommend** that you submit an additional form variable of arbitrary name chosen by the merchant and random value which changes from transaction to transaction. This will be returned to the website along with all the iVeri Lite form variables and all you need to do is check for the existence of the form variable and that the value thereof is the same as that generated and used during the submission of the iVeri Lite page.

9. APPENDIX D: AUTHORISATION INFORMATION

Note: This is just an extra step that can be used by the Merchant to make sure about the Transaction results. We would recommend that a Merchant should use this as a call to make sure the results he received back from the Authorise of a transaction are consistent with the results returned here. The URL to post to is defined in (section 7.2)

General descriptor	Name	length	values	In step 3 or step 6 of illustration 1	Notes
iVeri Application id	Lite_Merchant_ApplicationId	36		3 and 6	Allocated to the Merchant during registration
MerchantTrace	Lite_Merchant_Trace	64		3 and 6	Unique merchant identification for the request. This value is to be used by the merchant to confirm the status of the transaction.

The response to the Authorisation Information request is the following:

- Lite_Merchant_ApplicationId
- Lite_Merchant_Trace
- Ecom_Payment_Card_Number
- Ecom_Payment_Card_ExpDate_Month
- Ecom_Payment_Card_ExpDate_Year
- MerchantReference
- Lite_Order_Amount
- Lite_Order_BudgetPeriod
- Lite_Order_Terminal
- Lite_Order_AuthorisationCode
- Lite_BankReference
- Lite_TransactionDate
- Lite_TransactionIndex
- Lite_Payment_Card_Status
- Lite_Result_Description

10. TOKENISATION: TRANSACTIONINDEX ON SUBSEQUENT TRANSACTIONS

This section explains how to implement a follow up/subsequent transaction using the TransactionIndex returned from an initial/previous transaction processed successfully.

Merchants that wish to accept payments from regular customers without worrying about PCI DSS burdens of storing or retaining the card number have an option of submitting a unique identifier associated with the customers card number from a previously successfully processed transaction. In iVeri's realm, the identifier which the merchant can pass on subsequent transactions is called the **"TransactionIndex"**. This variable is an iVeri Gateway generated identifier commonly found in Gateway responses to the merchant

10.1 Requirement: Initial Transaction

When a merchant sends a transaction request(POST) to the iVeri Gateway, the response returned to the merchant generally contains a number of variables, some of which are important when performing subsequent transactions, in order for the merchant to implement subsequent transactions the following variables must be stored on the merchants database.

- Lite_TransactionIndex
- Ecom_Payment_Card_Number
- Ecom_Payment_Card_ExpDate_Month
- Ecom_Payment_Card_ExpDate_Year

10.2 Subsequent Transactions

When TransactionIndex is used on subsequent transactions, regular customers do not have to re-supply the card data details however this ONLY works if the merchant developer has made provisions for the following:

- Ability for the merchant to identify the customer, usually by means of user sign-in
- The merchant has successfully processed a transaction on the customers card at some point in time using the iVeri Gateway
- That the customer's profile is mapped to the correct Tokenised card details (TransactionIndex, Expiry date etc) returned on the initial or previously processed transaction

Once the above provisions are made, the merchant developer would be able to display to the cardholder a masked/dotted card number and the expiry date of the card. The cardholder would simply select the masked card on file and make a payment.

10.3 Subsequent Transactions implementation

On the request to the Gateway, the merchant developer has to pass all the required variables that pertain to a subsequent transaction so as to make sure that the iVeri Gateway handles the request appropriately.

Variables that should be present in the request to the iVeri Gateway are as follows:

- The **Lite_PanFormat** must be set to TransactionIndex
- Set the **Lite_TransactionIndex** to the actual TransactionIndex value e.g {000000-000000-000-0000 0000000}
- Set the **Ecom_Payment_Card_Number** to the Dotted PAN value

***Note**: These variables are defined in 8.2 iVeri specification. In addition to these the merchant will still need to pass all the other required variables.*

11. MERCHANT HASH TOKEN GENERATION

Then merchant token generation is a security measure introduced to hash merchant and transaction specific data elements, using SHA256 hashing algorithm. An effort which reduces the risk of data being exposed or intercepted by 3rd parties during the submission of transaction requests to the gateway.

11.1 Prerequisites

The below parameters must be set in the merchant Backoffice

- Enable **Token Verification** to "Yes" (By default this parameter is set to "No")
- Populate the **Lite Shared Secret** key parameter. Maximum length 32 characters (alpha-numeric)

11.2 Requirements

Merchants must generate the token on their web server and pass the generated token to the transaction request. The generated token must encompass the following data elements:

- **Lite_Order_Amount:** Total amount of the order as specified in 8.2
- **Lite_Merchant_ApplicationId** – The merchants app ID as specified in 8.2
- **Ecom_BillTo_Online_Email** – Email of the cardholder as specified in 8.2
- **TimeStamp** – The timestamp when the token is generated

NB The merchant will still need to pass all the required variables as defined in 8.2.

11.3 Token Verification Logic in the Hosted Payment Page

- If **Enable Token Verification** has been set to YES and **Lite Shared Secret** has not been set an exception will be thrown when submitting a transaction request
- If **Enable Token Verification** has been set to YES and **Lite_Transaction_Token** has not been set an exception will be thrown when submitting a transaction request
- If **Enable Token Verification** has been set to YES and the **Lite_Transaction_Token** does not match the calculated token an exception will be thrown

12. APPENDIX E: MERCHANT HASH TOKEN GENERATION

```
/*
*secretKey - Lite Shared Secret
*resource - /Lite/Authorise.aspx
*applicationId - Lite Application Id
*amount - Lite Order Amount
*emailAddress - Ecom BillTo Online Email
*/
public static string GenerateTransactionToken(string secretKey, string resource, string applicationId, string amount, string emailAddress)
{
 string time = Convert.ToString(UnixTimeStampUTC());
string token = secretKey + time + resource + applicationId + amount + emailAddress;
return String.Concat("x:", time, ":" + GetHashSha256(token));
}
public static Int32 UnixTimeStampUTC()
{
Int32 unixTimeStamp;
DateTime currentTime = DateTime.Now;
DateTime zuluTime = currentTime.ToUniversalTime();
DateTime unixEpoch = new DateTime(1970, 1, 1);
unixTimeStamp = (Int32)(zuluTime.Subtract(unixEpoch)).TotalSeconds;
return unixTimeStamp;
}
public static string GetHashSha256(string text)
{
```

```
byte[] bytes = Encoding.ASCII.GetBytes(text);
SHA256Managed hashstring = new SHA256Managed ();
byte[] hash = hashstring.ComputeHash(bytes);
string hashString = string.Empty;
foreach (byte x in hash)
 {
hashString += String.Format("{0:x2}", x);
 }
return hashString;
```

Sample Code:

ApplicationId:{c0f9f3e2-b75c-4864-b6c6-df1372fbedb0}
LiteSharedSecret:AFcWxV2NG9W4
Lite_Order_mound:4130
OnlineBilltoEmail:jdoe@mail.com
Timestamp:1471358394
Lite Transaction Token:
1471358394:8dd2f3c016662ef48743cf73a0b2e88de79b3e56791cbf48f8e0a688e363439915

13. LITEBOX – HOSTED PAYMENT PAGE

The LiteBox hosted payment page is an e-commerce solution that allows merchants to connect, send payment requests to the iVeri Payment Gateway without redirecting the cardholder away from their website. When implemented, the Litebox pops up and sits in-front of the merchant's website. From a merchant/cardholder point of view the LiteBox solution provides a more seamless checkout experience.

13.1 Integration Method

Merchants developers can download the Javascript library that handles the events to generate the button. The libray can be found on the following URL's

South African, Lesotho, Swaziland & Namibian merchants:

https://portal.nedsecure.co.za/scripts/jquery/js/jquery.litebox.js

Kenya & Zimbabwe merchants:

https://portal.host.iveri.com/scripts/jquery/js/jquery.litebox.js

CIM

https:// portal.merchant.cim.mu//scripts/jquery/js/jquery.litebox.js

Other:

https://portal.cscacquiring.com/scripts/jquery/js/jquery.litebox.js

13.2 Getting started

Step One: In order for the LiteBox to render correctly to the device being used and to connect to the appropriate gateway end point the below must be defined on the head tag

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<link rel="stylesheet" href="https://endpoint/scripts/bootstrap/css/bootstrap.min.css" />
<script src="https://endpoint/scripts/jquery/js/jquery.min.js"></script>
<script src="https://endpoint/scripts/bootstrap/js/bootstrap.min.js"></script>
<script src="https://endpoint/scripts/jquery/js/jquery.litebox.js"></script>
```

NB: replace "endpoint" with the appropriate gateway address

Step Two: Place the LiteBox button to be used to trigger the LiteBox page in the body tag, the merchant developer can set the color, and the label of their own choosing for the button. Download the Javascript library that handles the button on

https://endpoint/scripts/jquery/js/jquery.litebox.js . Also, to note is that the jQuery uses an id attribute on the HTML elements.

```
<!-- Button HTML -->
  <a id="iveri-litebox-button">Make Payment</a>
  <!-- Modal HTML -->
  <div id="iveri-litebox"></div>
```

Step Three: Initialize the Litebox. Define event handlers for the data that is to be returned from the gateway and when the Litebox closes.

```
<script>
      liteboxInitialise(portalUrl);          // pass the appropriate gateway address or endpoint
      function liteboxComplete(data) {
      /Place your code to handle the response here
      }
</script>
```

Step Four: Data returned to the merchant is JSON formatted

The following parameters are some of the data elements returned to the merchant, some of which will only be returned if present in the request message. Data parameters that can be passed to the iVeri gateway as part of the transaction request can be referenced in 8.2 **iVeri Lite Specification**

- Lite_Merchant_ApplicationId
- Lite_Merchant_Trace
- Ecom_Payment_Card_Number
- com_Payment_Card_ExpDate_Month
- Ecom_Payment_Card_ExpDate_Year
- MerchantReference
- Lite_Order_Amount
- Lite_Order_Terminal
- Lite_Order_AuthorisationCode
- Lite_BankReference
- Lite_TransactionDate
- Lite_TransactionIndex
- Lite_Payment_Card_Status
- Lite_Result_Description