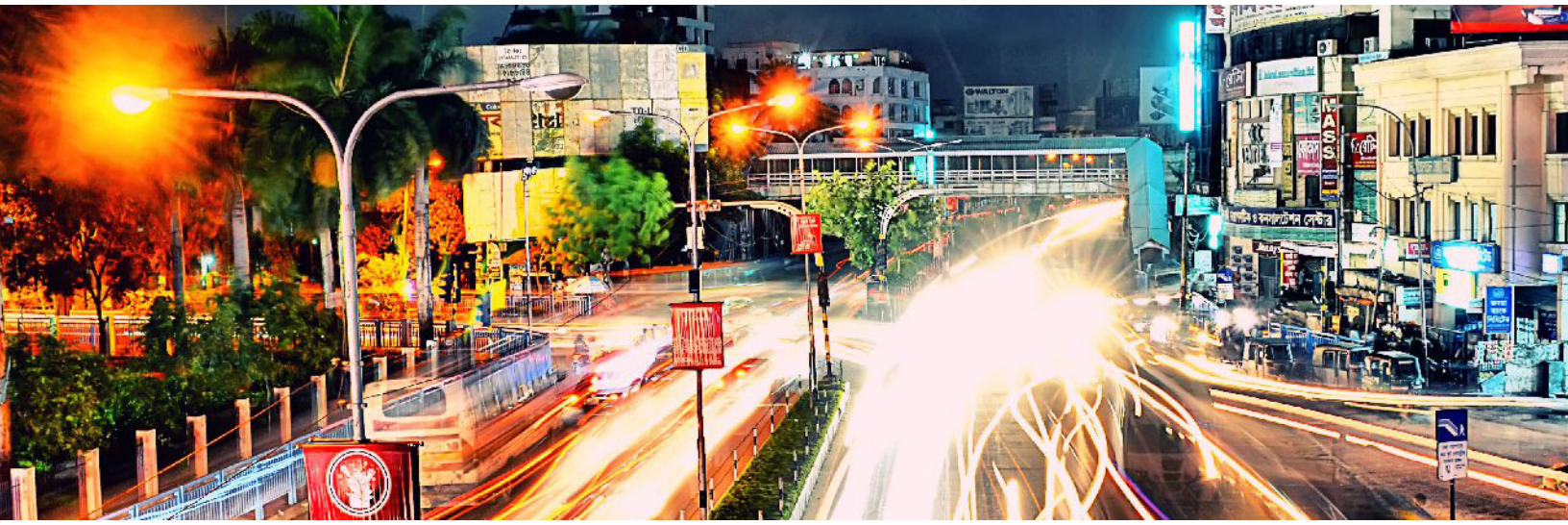
Samsung Pay

Using the SCMP API



cybersource
A Visa Solution

Cybersource Contact Information

For general information about our company, products, and services, go to http://www.cybersource.com.

For sales questions about any Cybersource service, email sales@cybersource.com or call 650-432-7350 or 888-330-2300 (toll free in the United States).

For support information about any Cybersource service, visit the Support Center: http://www.cybersource.com/support

Copyright

Restricted Rights Legends

Trademarks

Revision: September 2020

Contents

Recent Revisions to This Document

Release	Changes
September 2020	Change the name of *CyberSource through VisaNet* to *Visa Platform Connect*.
May 2020	Updated information about recurring payments. See "Supported Processors, Acquirers, Card Types, and Optional Features," page 9.
January 2020	Changed *payment network tokenization* to *authorizations with payment network tokens* throughout. Updated the Business Center procedure. See "Registering with Cybersource," page 12.
September 2019	This revision contains only editorial changes and no technical updates.
June 2019	Changed the name of *Mastercard SecureCode* to *Mastercard Identity Check*.
May 2019	This revision contains only editorial changes and no technical updates.

About This Guide

Audience and Purpose

This document is written for merchants who want to enable customers to use Samsung Pay to pay for in-app purchases. This document provides an overview of integrating the Samsung Pay SDK and describes how to request the Cybersource API to process an authorization. See "Using the Samsung Pay SDK," page 15, and "Authorizing a Payment," page 20. Merchants must use the Samsung Pay SDK to receive the customer's encrypted payment data before requesting the Cybersource API to process the transaction.

Conventions

Notes and Important Statements

A *Note* contains helpful suggestions or references to material not contained in the document.

An *Important* statement contains information essential to successfully completing a task or learning a concept.

Text and Command Conventions

Convention	Usage
bold	▪ Field and service names in text; for example: Include the **customer_cc_number** field. ▪ Items that you are instructed to act upon; for example: Click **Save**.
`Screen text`	Code examples and samples.

Related Documents

Cybersource Documents:

- *Getting Started with Cybersource Advanced for the SCMP API* (PDF | HTML)
- Cybersource SCMP API Client Developer Guide
- *Credit Card Services Using the SCMP API* (PDF | HTML)
- *Authorizations with Payment Network Tokens Using the SCMP API* (PDF | HTML)

Samsung Pay documents:

Samsung Pay Partner Portal

Refer to the Support Center for complete Cybersource technical documentation:

https://www.cybersource.com/en-us/support/technical-documentation.html

Customer Support

For support information about any Cybersource service, visit the Support Center:

http://www.cybersource.com/support

CHAPTER 1

Introduction

Requirements

Samsung Pay relies on payment network tokens. You can sign up for Samsung Pay only when both of the following statements are true:

- Your processor supports payment network tokens.
- Cybersource supports payment network tokens with your processor.

When one or both of the preceding statements are not true, you must take one of the following actions before you can sign up for Samsung Pay:

- Obtain a new merchant account with a processor that supports payment network tokens.
- Wait until your processor supports payment network tokens.

You must create:

- A Cybersource account. If you do not already have a Cybersource account, contact your local Cybersource sales representative:

  http://www.cybersource.com/locations/

- A merchant account with a supported processor. See "Supported Processors, Acquirers, Card Types, and Optional Features," page 9.

- A profile on the Samsung Pay Partner Portal, and you must obtain a partner ID. See "Registration," page 11.

Supported Processors, Acquirers, Card Types, and Optional Features

Merchant-initiated transactions, multiple partial captures, and subsequent authorizations are described in *Authorizations with Payment Network Tokens Using the SCMP API* (PDF | HTML). Recurring payments and split shipments are described in *Credit Card Services Using the SCMP API* (PDF | HTML).

Table 1 Supported Processors, Card Types, and Optional Features

Processors	Card Types	Optional
American Express Direct	American Express	■ Multiple partial captures ■ Recurring payments
Barclays	Visa, Mastercard, JCB, Maestro (International), Maestro (UK Domestic) If you support Maestro (UK Domestic), you must also support Maestro (International), and you must support Mastercard Identity Check for both card types.	■ Multiple partial captures ■ Recurring payments
Chase Paymentech Solutions	Visa, Mastercard, American Express, Discover, Diners Club, JCB, Carte Blanche, Maestro (International)	■ Multiple partial captures ■ Recurring payments
Elavon Americas	Visa, Mastercard, American Express, JCB, Diners Club, Discover, China UnionPay	■ Merchant-Initiated transactions ■ Multiple partial captures ■ Recurring payments
FDC Compass	Visa, Mastercard, American Express, Discover, Diners Club, JCB	■ Multiple partial captures ■ Recurring payments
FDC Nashville Global	Visa, Mastercard, American Express, Discover, Diners Club, JCB, China UnionPay	■ Multiple partial captures ■ Recurring payments
GPN	Visa, Mastercard, American Express, Discover, Diners Club, JCB	Split shipments

Table 1 Supported Processors, Card Types, and Optional Features (Continued)

Processors	Card Types	Optional
JCN Gateway	Visa, Mastercard, American Express, Diners Club, JCB, NICOS house card, ORICO house card	Multiple partial captures
OmniPay Direct ■ Bank of America Merchant Services ■ First Data Merchant Solutions (Europe) ■ Global Payments International Acquiring	Visa, Mastercard, Discover, Diners Club, Maestro (UK Domestic), Maestro (International)	■ Multiple partial captures ■ Recurring payments
Streamline	Visa, Mastercard	■ Multiple partial captures ■ Recurring payments ■ Subsequent authorizations
TSYS Acquiring Solutions	Visa, Mastercard, American Express	Multiple partial captures
Visa Platform Connect. The supported acquirer is: ■ Vantiv Visa Platform Connect is a single processor with multiple acquirers.	Visa, Mastercard, American Express, Discover, JCB, Diners Club	■ Recurring payments ■ Split shipments

Transaction Endpoints

Test transactions:

- Akamai endpoint: http://ics2testa.ic3.com
- Non-Akamai endpoint: http://ics2test.ic3.com/

Live transactions:

- Akamai endpoint: http://ics2a.ic3.com
- Non-Akamai endpoint: http://ics2.ic3.com/

CHAPTER 2

Registration

Registering with Samsung

To register with Samsung:

Step 1 Create a profile by completing the merchant application on the Samsung Pay Partner Portal. Samsung will contact you if clarification is needed.

Step 2 After your merchant application is approved, you receive a unique partner ID. Include this ID in your application.

> ⚠ You need the partner ID in order to generate the Certificate Signing Request (CSR) file in the Business Center. See "Registering with Cybersource," page 12. Samsung requires the CSR file in order to encrypt sensitive payment data; it contains an identifier and public key.

Step 3 Using the Samsung Pay Partner Portal, upload the CSR file.

Step 4 Enter an application name and a package name.

Step 5 When you associate the CSR file with the application, Samsung generates a product ID.

Step 6 Create login details for application developers on the Samsung Pay Partner Portal.

Step 7 Download and integrate the Samsung Pay SDK into your application. See "Using the Samsung Pay SDK," page 15.

The SDK contains:

- A Javadoc
- The Samsung Pay SDK files *samsungpay.jar* and *sdk-v1.0.0.jar*
- A sample app
- The branding guide
- Image files

Step 8 Register a Samsung account ID and request a *debug-api-key* file using the Samsung Pay Partner Portal. The *debug-api-key* file is valid for three months. See "Using the API Key," page 14.

The Samsung account ID, the *debug-api-key*, and the product ID are used to validate your application so that you can use the Samsung Pay SDK for testing.

Step 9 Submit your application for approval using the Samsung Pay Partner Portal. Upload the final version of the Android Application Package (APK) file using the Samsung Pay Partner Portal, and include screenshots of your checkout page displaying the Samsung Pay logo.

Registering with Cybersource

To register with Cybersource:

Step 1 Log in to the Business Center:

- Create a CSR file for test transactions: https://ebctest.cybersource.com/ebc2/
- Create a CSR file for live transactions: https://ebc2.cybersource.com/ebc2/

Step 2 On the left navigation pane, click the **Payment Configuration** icon.

Step 3 Click **Digital Payment Solutions**. The Digital Payments page appears.

Step 4 Click **Configure**. The Samsung Pay Registration panel opens.

Step 5 Enter your Samsung partner ID.

Step 6 Click Generate New CSR.

Step 7 To download your CSR, click the **Download** icon next to the key.

Step 8 Follow your browser's instructions to save and open the file.

Only one CSR is permitted for each unique Samsung partner ID. If you modify your Samsung partner ID you must generate a new CSR.

Step 9 Complete the enrollment process by submitting your CSR to Samsung.

CHAPTER 3

Integrating the Samsung SDK

Creating a Project

To create a project using Android Studio:

Step 1 Download Android Studio.

Step 2 Open Android Studio and click **Start a new Android Studio project**.

Step 3 In the New Project settings, enter the name of your application and the company domain.

Step 4 To change the package name, click **Edit**. By default, Android Studio sets the last element of the project's package name to the name of your application.

Step 5 Click **Next**.

Step 6 In the Target Android Devices settings, choose the required API levels.

Step 7 Click **Next**.

Step 8 Choose the required activity and click **Finish**.

Integrating the Samsung Pay SDK

To integrate the Samsung Pay SDK:

Step 1 Add the *samsungpay.jar* and *sdk-v1.0.0.jar* files to the *libs* folder of your Android project.

Step 2 Choose **Gradle Scripts > build.gradle** and enter the dependencies shown below.

```
    dependencies {
       compile files('libs/samsungpay.jar')
       compile files('libs/sdk-v1.0.0.jar')
}
```

Step 3 Import the package.

```
import com.samsung.android.sdk.samsungpay;
```

Using the API Key

The API key is used to verify that your app (in debug mode or release mode) can use the Samsung Pay SDK APIs with the Samsung Pay application. To get the API key, you must create a *debug-api-key* file (Step 8 in "Registering with Samsung," page 11) and include it in the *manifest* file.

To use the API key:

Step 1 Include the API key in the *manifest* file with a custom tag. This enables the merchant app android *manifest* file to provide the `DebugMode`, `spay_debug_api_key` values as meta-data.

Example 1 Debug Mode

```
<meta-data
    android:name="debug_mode"
    android:value="Y" />
<meta-data
    android:name="spay_debug_api_key"
    android:value="asdfggkndkeie17283094858" />
```

Example 2 Release Mode

```
<meta-data
    android:name="debug_mode"
    android:value="N" />
```

CHAPTER 4

Using the Samsung Pay SDK

Eligibility

Initialize the `SSamsungPay` class to verify that your application is eligible for Samsung Pay and to display the Samsung Pay button to the customer (refer to branding guidelines).

The `SSamsungPay` class provides the following API methods:

- `initialize()`—initializes the Samsung Pay SDK and verifies eligibility for Samsung Pay, including the device, software, and business area.

  Request the `initialize()` API method of the `SSamsungPay` class before using the Samsung Pay SDK.

- `getVersionCode()`—retrieves the version number of the Samsung Pay SDK as an integer.
- `getVersionName()`—retrieves the version name of the Samsung Pay SDK as a string.

After the `initialize()` API method request is successful, display the Samsung Pay button to the customer.

If the `initialize()` API method request fails, the method displays a `SsdkUnsupportedException` or `NullPointerException` error.

- `SsdkUnsupportedException`—the device is not a Samsung device or does not support the Samsung Pay package.
- `NullPointerException`—the context passed is null.

Example 3 Samsung Pay Class

```
SSamsungPay spay = new SSamsungPay();
try {
    spay.initialize(mContext);
} catch (SsdkUnsupportedException e1) {
    e1.printStackTrace();
    pay_button.setVisibility(View.INVISIBLE);
}
```

Payment Request

Initiating a Payment

To initiate a payment:

Step 1 Include the following fields in the `PaymentInfo` class:

If the required fields are not included, you receive a `NullPointerException` error.

- Merchant Name—the merchant name as it appears on the payment sheet of Samsung Pay and customer's bank statement. This field is required.
- Amount—this field is required.
- Payment Protocol—3D Secure. This field is required.
- Permitted Card Brands—specify the card brands that are supported such as Visa, Mastercard, or American Express. This field is required.
- Merchant ID
- Order Number
- Shipping Address—this field is required if SEND_SHIPPING or NEED_BILLING_AND_SEND_SHIPPING is set for `AddressVisibilityOption`.
- Address Visibility Option
- Card Holder Name
- Recurring Option

Example 4 Transaction Request Structure

```
private PaymentInfo makeTransactionDetails() {
// Supported card brands
ArrayList<CardInfo.Brand> brandList = new ArrayList<CardInfo.Brand>();
if (visaBrand.isChecked())
brandList.add(CardInfo.Brand.VISA);
if (mcBrand.isChecked())
brandList.add(CardInfo.Brand.Mastercard);
if (amexBrand.isChecked())
brandList.add(CardInfo.Brand.AMERICANEXPRESS);

// Basic payment information
PaymentInfo paymentReq = new PaymentInfo.Builder()
.setMerchantId("merchantID")
.setMerchantName("Test").setAmount(getAmount())
.setShippingAddress(getShippingAddressInfo())
.setOrderNumber(orderNoView.getText().toString())
.setPaymentProtocol(PaymentProtocol.PROTOCOL_3DS)
.setAddressInPaymentSheet(AddressInPaymentSheet.DO_NOT_SHOW)
.setAllowedCardBrands(brandList) .setRecurringEnabled(isRecurring)
.setCardHolderNameEnabled(isCardHolderNameRequired)
.build();
return paymentReq;
}

// Add shipping address details
private Address getShippingAddressInfo() {
Address address = new Address.Builder()
.setAddressee(name.getText().toString())
.setAddressLine1(addLine1.getText().toString())
.setAddressLine2(addline2.getText().toString())
.setCity(city.getText().toString())
.setState(state.getText().toString())
.setCountryCode(country.getSelectedItem().toString())
.setPostalCode(zip.getText().toString()).build(); return address;
}

// Add amount details private Amount getAmount() {
Amount amount = new Amount.Builder()
.setCurrencyCode(currencyType.getSelectedItem().toString())
.setItemTotalPrice(productPrice.getText().toString())
.setShippingPrice(shippingPrice.getText().toString())
.setTax(taxPrice.getText().toString())
.setTotalPrice(totalAmount.getText().toString()).build();
return amount;
}
```

Requesting a Payment

To request a payment:

Step 1 Use the `startSamsungPay()` API method in the `PaymentManager` class.

The `PaymentManager` class includes the following API methods:

- `startSamsungPay()`—requests to initiate payment with Samsung Pay.
- `updateAmount()`—updates the transaction amount if shipping address or card information is updated by Samsung Pay.
- `updateAmountFailed()`—returns an error code when the new amount cannot be updated because of a wrong address.

Step 2 Request the `startSamsungPay()` API method and include the following data:

- `PaymentInfo`—contains payment information.
- `PID`—the product ID created in the Samsung Pay Partner Portal. See "Registration," page 11.
- `StatusListener`—the result of the payment request is delivered to `StatusListener`. This listener should be registered before you call the `startSamsungPay()` API method.

When you request the `startSamsungPay()` API method, the Samsung Pay online payment sheet is displayed on your application. The customer selects a registered card for payment and can also update the billing and shipping address.

The payment reply is delivered as one of the following events to `StatusListener`:

- `onSuccess()`—this event is requested when Samsung Pay confirms the payment. It includes `encryptedPaymentCredential` in JSON format. See Table 2, "Encrypted Payment Credential," on page 19.
- `onFailure()`—this event is requested when the transaction fails. It returns an error code and error message.

Example 5 Request startSamsungPay() API Method

```
public void onPayButtonClicked(View v) {
    // Call startSamsungPay() method of PaymentManager class.
    // To create a transaction request for makeTransactionDetails() in
    the following code, see Example 4, "Transaction Request Structure,"
    on page 17.
    try {
        mPaymentManager.startSamsungPay(makeTransactionDetails(), "enter
        product ID",
mStatusListener);
    } catch (NullPointerException e) {
    e.printStackTrace();
    }
}

private PaymentManager.StatusListener mStatusListener = new
PaymentManager.StatusListener() {
    @Override
    public void onFailure(int errCode, String msg) {
        Log.d(TAG, " onFailed );
    }
    @Override
    public void onSuccess(PaymentInfo arg0, String result) {
        Log.d(TAG, "onSuccess ");
    };
```

Table 2 Encrypted Payment Credential

Payment Credential	Description
method	Payment protocol: 3D Secure.
merchant_ref	Merchant reference code.
billing_address.street	Number, street name.
billing_address.state_province	Two-letter state code.
billing_address.zip_postal_code	Five-character zip code.
billing_address.city	City name.
billing_address.county	Two letter country code.
3ds.type	S for Samsung Pay. Encrypted.
3ds.version	Current version 100. Encrypted.
3ds.data	Base64-encoded payment data. Encrypted.

For information on how to decrypt the encrypted payment credential, see:

https://pay.samsung.com/developers

CHAPTER 5

Authorizing a Payment

Your payment processor can include API reply fields that are not documented in this guide. See *Credit Card Services Using the SCMP API* (PDF | HTML) for detailed descriptions of additional API reply fields.

Merchant Decryption

Visa Transaction

See "Request Fields," page 36, and "Reply Fields," page 43, for detailed field descriptions.

To request an authorization for a Visa transaction:

Step 1 Set the **customer_cc_number** field to the payment network token value.

Step 2 Set the **customer_cc_expmo** and **customer_cc_expyr** fields to the payment network token expiration date values.

Step 3 Set the **cavv** field to the 3D Secure cryptogram of the payment network token.

Step 4 Set the **network_token_cryptogram** field to the network token cryptogram.

Step 5 Set the **payment_network_token_transaction_type** field to `1`.

Step 6 Set the **e_commerce_indicator** field to `internet`.

Step 7 Set the **payment_solution** field to `008`.

Example 6 Merchant Decryption Authorization Request (Visa)

```
bill_address1=111 S. Division St.
bill_address2=Suite 123
bill_city=Ann Arbor
bill_country=US
bill_state=MI
bill_zip=48104-2201
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2021
customer_cc_number=xxxx100000000xxxx
customer_email=demo@example.com
customer_firstname=James
customer_ipaddress=66.123.123.2
customer_lastname=Smith
customer_phone=999-999-9999
e_commerce_indicator=internet
grand_total_amount=100.00
ics_applications=ics_auth
merchant_id=demomerchant
merchant_ref_number=demorefnum
cavv=ABCDEFabcdefABCDEFabcdef0987654321234567
payment_network_token_transaction_type=1
solution_type=008
```

Example 7 Merchant Decryption Authorization Reply (Visa)

```
request_token=Ahj/7wSR5C/kX63O2hAKIkGLNkwcsmrSHHlU5tGHRT/hHgzc8BT/hHgk
currency=usd
request_id=4465837560045000001541
auth_rflag=SOK
ics_rmsg=Request was processed successfully.
auth_auth_amount=100.00
auth_rcode=1
auth_trans_ref_no=13209254CGJSMQCQ
auth_auth_code=888888
auth_rmsg=Request was processed successfully.
ics_rflag=SOK
auth_auth_response=100
auth_avs_raw=I1
auth_auth_time=2015-11-03T204917Z
merchant_ref_number=demorefnum
ics_rcode=1
```

Mastercard Transaction

See "Request Fields," page 36, and "Reply Fields," page 43, for detailed field descriptions.

To request an authorization for a Mastercard transaction:

Step 1 Set the **customer_cc_number** field to the payment network token value.

Step 2 Set the **customer_cc_expmo** and **customer_cc_expyr** fields to the payment network token expiration date values.

Step 3 Set the **ucaf_authentication_data** field to the 3D Secure cryptogram of the payment network token.

Step 4 Set the **network_token_cryptogram** field to the network token cryptogram.

Step 5 Set the **ucaf_collection_indicator** field to `2`.

Step 6 Set the **payment_network_token_transaction_type** field to `1`.

Step 7 Set the **e_commerce_indicator** field to `spa`.

Step 8 Set the **payment_solution** field to `008`.

Example 8 Merchant Decryption Authorization Request (Mastercard)

```
bill_address1=111 S. Division St.
bill_address2=Suite 123
bill_city=Ann Arbor
bill_country=US
bill_state=MI
bill_zip=48104-2201
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2021
customer_cc_number=xxxx55555555xxxx
customer_email=demo@example.com
customer_firstname=James
customer_ipaddress=66.123.123.2
customer_lastname=Smith
customer_phone=999-999-9999
grand_total_amount=100.00
ics_applications=ics_auth
merchant_id=demomerchant
merchant_ref_number=demorefnum
ucaf_authentication_data=ABCDEFabcdefABCDEFabcdef0987654321234567
ucaf_collection_indicator=2
payment_network_token_transaction_type=1
solution_type=008
```

Example 9 Merchant Decryption Authorization Reply (Mastercard)

```
request_token=Ahj/7wSR5C/p6oJEy1gKIkGLNkwcsmrWHHlU5tGHST/hHgzdACT/hVB3c
currency=usd
request_id=4465838340055000001541
auth_rflag=SOK
ics_rmsg=Request was processed successfully.
auth_auth_amount=100.00
auth_rcode=1
auth_trans_ref_no=13209255CGJSMQCR
auth_auth_code=888888
auth_rmsg=Request was processed successfully.
ics_rflag=SOK
auth_auth_response=100
auth_avs_raw=I1
auth_auth_time=2015-11-03T205035Z
merchant_ref_number=demorefnum
ics_rcode=1
```

American Express Transaction

See "Request Fields," page 36, and "Reply Fields," page 43, for detailed field descriptions.

To request an authorization for an American Express transaction:

Step 1 Set the **customer_cc_number** field to the payment network token value.

Step 2 Set the **customer_cc_expmo** and **customer_cc_expyr** fields to the payment network token expiration date values.

Step 3 Set the **cavv** field to the 3D Secure cryptogram of the payment network token.

Include the whole 20-byte cryptogram in the **cavv** field. For a 40-byte cryptogram, split the cryptogram into two 20-byte binary values (block A and block B). Set the **cavv** field to the block A value and set the **xid** field to the block B value.

Step 4 Set the **network_token_cryptogram** field to the network token cryptogram.

Step 5 Set the **payment_network_token_transaction_type** field to `1`.

Step 6 Set the **e_commerce_indicator** field to `aesk`.

Step 7 Set the **payment_solution** field to `008`.

Example 10 Merchant Decryption Authorization Request (American Express)

```
bill_address1=111 S. Division St.
bill_address2=Suite 123
bill_city=Ann Arbor
bill_country=US
bill_state=MI
bill_zip=48104-2201
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2021
customer_cc_number=xxxx82246310xxxx
customer_email=demo@example.com
customer_firstname=James
customer_ipaddress=66.123.123.2
customer_lastname=Smith
customer_phone=999-999-9999
grand_total_amount=100.00
ics_applications=ics_auth
merchant_id=demomerchant
merchant_ref_number=demorefnum
cavv=ABCDEFabcdefABCDEFabcdef0987654321234567
xid=12345678909876543 21ABCDEFabcdefABCDEF123
payment_network_token_transaction_type=1
solution_type=008
```

Example 11 Merchant Decryption Authorization Reply (American Express)

```
request_token=Ahj/7wSR5C/wGXKw1xAKIkGLNkwcsmraHHlU5tGHaT/hHgzecDT/h6BBL
currency=usd
request_id=4465839210285000001541
auth_rflag=SOK
ics_rmsg=Request was processed successfully.
auth_auth_amount=100.00
auth_rcode=1
auth_trans_ref_no=13209256CGJSMQCZ
auth_auth_code=888888
auth_rmsg=Request was processed successfully.
ics_rflag=SOK
auth_auth_response=100
auth_avs_raw=I1
auth_auth_time=2015-11-03T205202Z
merchant_ref_number=demorefnum
ics_rcode=1
```

JCB Transaction

See "Request Fields," page 36, and "Reply Fields," page 43, for detailed field descriptions.

To request an authorization for a JCB transaction:

Step 1 Set the **customer_cc_number** field to the payment network token value.

Step 2 Set the **customer_cc_expmo** and **customer_cc_expyr** fields to the payment network token expiration date values.

Step 3 Set the **cavv** field to the 3D Secure cryptogram of the payment network token.

Step 4 Set the **network_token_cryptogram** field to the network token cryptogram.

Step 5 Set the **payment_network_token_transaction_type** field to `1`.

Step 6 Set the **eci_raw** field to the ECI value contained in the Samsung Pay reply message.

Step 7 Set the **payment_solution** field to `008`.

Example 12 Merchant Decryption Authorization Request (JCB)

```
bill_address1=123 Main Street
bill_address2=Suite 12345
bill_city=Small Town
bill_country=US
bill_state=CA
bill_zip=98765
card_type=007
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx11111111xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
eci_raw=05
grand_total_amount=100.00
ics_applications=ics_auth
merchant_id=med123
cavv=EHuWW9PiBkWvqE5juRwDzAUFBAk=
payment_network_token_transaction_type=1
payment_solution=008
```

Example 13 Merchant Decryption Authorization Reply (JCB)

```
auth_auth_amount=100.00
auth_auth_avs=X
auth_auth_code=888888
auth_auth_response=100
auth_avs_raw=I1
auth_rcode=1
auth_rflag=SOK
auth_rmsg=Request was processed successfully.
auth_trans_ref_no=15356268CR2XF23X
currency=USD
ics_rcode=1
ics_rflag=SOK
ics_rmsg=Request was processed successfully.
merchant_ref_number=ref123
request_id=469736926810612460154l
request_token=Ahj/7wSR/UoVm1bMmziHSZjMECT/h+KjMHSB04gwGA2dDjQoxQAAA6xdr
```

Cybersource Decryption

Visa Transaction

See "Request Fields," page 36, and "Reply Fields," page 43, for detailed field descriptions.

To request an authorization for a Visa transaction:

Step 1 Set the **encrypted_payment_data** field to the value that was returned from Samsung Pay in the *3ds.data* block.

a Retrieve the payment data from Samsung Pay in JSON Web Encryption (JWE) format.

b Encode it in Base64.

c Retrieve the corresponding Key ID (KID) with encryption and set the values as:

```
{
    "publicKeyHash": "kid",
    "version": "100",
    "data": "encoded data from step b above"
}
```

d Encode the structure in Base64.

e Add the value to the **encrypted_payment_data** field.

Step 2 Set the **encrypted_payment_descriptor** field to `RklEPUNPTU1PTi5TQU1TVU5HLklOQVBQLlBBWU1FTlQ=`.

Step 3 Set the **payment_network_token_transaction_type** field to `1`.

Step 4 Set the **e_commerce_indicator** field to `internet`.

Step 5 Set the **payment_solution** field to `008`.

Example 14 Cybersource Decryption Authorization Request (Visa)

```
bill_address1=111 S. Division St.
bill_address2=Suite 123
bill_city=Ann Arbor
bill_country=US
bill_state=MI
bill_zip=48104-2201
currency=usd
customer_email=demo@example.com
customer_firstname=James
customer_ipaddress=66.123.123.2
customer_lastname=Smith
customer_phone=999-999-9999
e_commerce_indicator=internet
grand_total_amount=100.00
ics_applications=ics_auth
merchant_id=demomerchant
merchant_ref_number=demorefnum
encrypted_payment_data=ABCDEFabcdefABCDEFabcdef0987654321234567
encrypted_payment_
descriptor=RklEPUNPTU1PTi5TQU1TVU5HLklOQVBQLlBBWU1FTlQ=
payment_network_transaction_type=1
solution_type=008
```

Example 15 Cybersource Decryption Authorization Reply (Visa)

```
request_token=Ahj/7wSR5C/kX63O2hAKIkGLNkwcsmrSHHlU5tGHRT/hHgzc8BT/hHgk
currency=usd
request_id=4465837560045000001541
auth_rflag=SOK
ics_rmsg=Request was processed successfully.
auth_auth_amount=100.00
auth_rcode=1
auth_trans_ref_no=13209254CGJSMQCQ
auth_auth_code=888888
auth_rmsg=Request was processed successfully.
ics_rflag=SOK
auth_auth_response=100
auth_avs_raw=I1
merchant_ref_number=demorefnum
ics_rcode=1
token_prefix=294672
token_suffix=4397
token_expirationMonth=08
token_expirationYear=2021
```

Mastercard Transaction

See "Request Fields," page 36, and "Reply Fields," page 43, for detailed field descriptions.

To request an authorization for a Mastercard transaction:

Step 1 Set the **encrypted_payment_data** field to the value that was returned from Samsung Pay in the *3ds.data* block.

a Retrieve the payment data from Samsung Pay in JSON Web Encryption (JWE) format.

b Encode it in Base64.

c Retrieve the corresponding Key ID (KID) with encryption and set the values as:

```
{
    "publicKeyHash": "kid",
    "version": "100",
    "data": "encoded data from step b above"
}
```

d Encode the structure in Base64.

e Add the value to the **encrypted_payment_data** field.

Step 2 Set the **encrypted_payment_descriptor** field to `RklEPUNPTU1PTi5TQU1TVU5HLklOQVBQLlBBWU1FTlQ=`.

Step 3 Set the **e_commerce_indicator** field to `spa`.

Step 4 Set the **payment_network_token_transaction_type** field to `1`.

Step 5 Set the **payment_solution** field to `008`.

Example 16 Cybersource Decryption Authorization Request (Mastercard)

```
bill_address1=111 S. Division St.
bill_address2=Suite 123
bill_city=Ann Arbor
bill_country=US
bill_state=MI
bill_zip=48104-2201
currency=usd
customer_email=demo@example.com
customer_firstname=James
customer_ipaddress=66.123.123.2
customer_lastname=Smith
customer_phone=999-999-9999
e_commerce_indicator=spa
grand_total_amount=100.00
ics_applications=ics_auth
merchant_id=demomerchant
merchant_ref_number=demorefnum
encrypted_payment_data=ABCDEFabcdefABCDEFabcdef0987654321234567
encrypted_payment_
descriptor=RklEPUNPTU1PTi5TQU1TVU5HLklOQVBQLlBBWU1FTlQ=
payment_network_transaction_type=1
solution_type=008
```

Example 17 Cybersource Decryption Authorization Reply (Mastercard)

```
request_token=Ahj/7wSR5C/p6oJEy1gKIkGLNkwcsmrWHHlU5tGHST/hHgzdACT/hVB3c
currency=usd
request_id=4465838340055000001541
auth_rflag=SOK
ics_rmsg=Request was processed successfully.
auth_auth_amount=100.00
auth_rcode=1
auth_trans_ref_no=13209255CGJSMQCR
auth_auth_code=888888
auth_rmsg=Request was processed successfully.
ics_rflag=SOK
auth_auth_response=100
auth_avs_raw=I1
auth_auth_time=2015-11-03T205035Z
merchant_ref_number=demorefnum
ics_rcode=1
token_prefix=128945
token_suffix=2398
token_expirationMonth=08
token_expirationYear=2021
```

American Express Transaction

See "Request Fields," page 36, and "Reply Fields," page 43, for detailed field descriptions.

To request an authorization for an American Express transaction:

Step 1 Set the **encrypted_payment_data** field to the value that was returned from Samsung Pay in the *3ds.data* block.

a Retrieve the payment data from Samsung Pay in JSON Web Encryption (JWE) format.

b Encode it in Base64.

c Retrieve the corresponding Key ID (KID) with encryption and set the values as:

```
{
    "publicKeyHash": "kid",
    "version": "100",
    "data": "encoded data from step b above"
}
```

d Encode the structure in Base64.

e Add the value to the **encrypted_payment_data** field.

Step 2 Set the **encrypted_payment_descriptor** field to `RklEPUNPTU1PTi5TQU1TVU5HLklOQVBQLlBBWU1FTlQ=`.

Step 3 Set the **payment_network_token_transaction_type** field to `1`.

Step 4 Set the **e_commerce_indicator** field to `aesk`.

Step 5 Set the **payment_solution** field to `008`.

Example 18 Cybersource Decryption Authorization Request (American Express)

```
bill_address1=111 S. Division St.
bill_address2=Suite 123
bill_city=Ann Arbor
bill_country=US
bill_state=MI
bill_zip=48104-2201
card_type=003
currency=usd
customer_email=demo@example.com
customer_firstname=James
customer_ipaddress=66.123.123.2
customer_lastname=Smith
customer_phone=999-999-9999
e_commerce_indicator=aesk
grand_total_amount=100.00
ics_applications=ics_auth
merchant_id=demomerchant
merchant_ref_number=demorefnum
encrypted_payment_data=ABCDEFabcdefABCDEFabcdef0987654321234567
encrypted_payment_
descriptor=RklEPUNPTU1PTi5TQU1TVU5HLklOQVBQLlBBWU1FTlQ=
payment_network_transaction_type=1
solution_type=008
```

Example 19 Cybersource Decryption Authorization Reply (American Express)

```
request_token=Ahj/7wSR5C/wGXKw1xAKIkGLNkwcsmraHHlU5tGHaT/hHgzecDT/h6BBL
currency=usd
request_id=4465839210285000001541
auth_rflag=SOK
ics_rmsg=Request was processed successfully.
auth_auth_amount=100.00
auth_rcode=1
auth_trans_ref_no=13209256CGJSMQCZ
auth_auth_code=888888
auth_rmsg=Request was processed successfully.
ics_rflag=SOK
auth_auth_response=100
auth_avs_raw=I1
auth_auth_time=2015-11-03T205202Z
merchant_ref_number=demorefnum
ics_rcode=1
token_prefix=593056
token_suffix=0842
token_expirationMonth=08
token_expirationYear=2021
```

JCB Transaction

See "Request Fields," page 36, and "Reply Fields," page 43, for detailed field descriptions.

To request an authorization for a JCB transaction:

Step 1 Set the **encrypted_payment_data** field to the Base64-encoded value obtained from the **paymentData** property of the **PKPaymentToken** object.

Step 2 Set the **encrypted_payment_descriptor** field to `RklEPUNPTU1PTi5TQU1TVU5HLklOQVBQLlBBWU1FTlQ=`.

Step 3 Set the **payment_solution** field to `008`.

Example 20 Cybersource Decryption Authorization Request (JCB)

```
bill_address1=123 Main Street
bill_address2=Suite 12345
bill_city=Small Town
bill_country=US
bill_state=CA
bill_zip=98765
card_type=007
currency=usd
customer_cc_expmo=12
customer_cc_expyr=2031
customer_cc_number=xxxx55555555xxxx
customer_email=js@example.com
customer_firstname=Jane
customer_lastname=Smith
customer_phone=999-999-9999
eci_raw=05
grand_total_amount=100.00
ics_applications=ics_auth
merchant_id=med123
cavv=EHuWW9PiBkWvqE5juRwDzAUFBAk=
payment_network_token_transaction_type=1
payment_solution=008
```

Example 21 Cybersource Decryption Authorization Reply (JCB)

```
auth_auth_amount=100.00
auth_auth_avs=X
auth_auth_code=888888
auth_auth_response=100
auth_avs_raw=I1
auth_rcode=1
auth_rflag=SOK
auth_rmsg=Request was processed successfully.
auth_trans_ref_no=15356268CR2XF23X
currency=USD
ics_rcode=1
ics_rflag=SOK
ics_rmsg=Request was processed successfully.
merchant_ref_number=ref123
request_id=4697369268106124601541
request_token=Ahj/7wSR/UoVm1bMmziHSZjMECT/h+KjMHSB04gwGA2dDjQoxQAAA6xdr
```

Additional Cybersource Services

Refer to *Credit Card Services Using the SCMP API* (PDF | HTML) for information on how to request these follow-on services.

Table 3 Cybersource Services

Cybersource Service	Description
Capture	A follow-on service that uses the request ID returned from the previous authorization. The request ID links the capture to the authorization. This service transfers funds from the customer's account to your bank and usually takes two to four days to complete.
Sale	A sale is a bundled authorization and capture. Request the authorization and capture services at the same time. Cybersource processes the capture immediately.
Authorization Reversal	A follow-on service that uses the request ID returned from the previous authorization. An authorization reversal releases the hold that the authorization placed on the customer's credit card funds. Use this service to reverse an unnecessary or undesired authorization.

APPENDIX A

API Fields

Data Type Definitions

For more information about these data types, see the World Wide Web Consortium (W3C) XML Schema Part 2: Datatypes Second Edition.

Table 4 Data Type Definitions

Data Type	Description
Date and time	Format is YYYY-MM-DDThhmmssZ, where: ■ T separates the date and the time ■ Z indicates Coordinated Universal Time (UTC), also known as Greenwich Mean Time (GMT) **Example** 2020-01-11T224757Z equals January 11, 2020, at 22:47:57 (10:47:57 p.m.)
Decimal	Number that includes a decimal point **Example** 23.45, -0.1, 4.0, 90809.0468
Integer	Whole number {..., -3, -2, -1, 0, 1, 2, 3, ...}
Nonnegative integer	Whole number greater than or equal to zero {0, 1, 2, 3, ...}
Positive integer	Whole number greater than zero {1, 2, 3, ...}
String	Sequence of letters, numbers, spaces, and special characters

Relaxed Requirements for Address Data

To enable relaxed requirements for address data and expiration date, contact Cybersource Customer Support to have your account configured for this feature. For details about relaxed requirements, see the Relaxed Requirements for Address Data and Expiration Date page.

Request Fields

Unless otherwise noted, all field names are case sensitive and all fields accept special characters such as `@`, `#`, and `%`.

Table 5 Request Fields

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
bill_address1	First line of the billing street address. **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting. See "Relaxed Requirements for Address Data," page 35.	ics_auth (See description)	String (60)
bill_address2	Additional address information. **Example** Attention: Accounts Payable	ics_auth (R)	String (60)
bill_city	City of the billing address. **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting. See "Relaxed Requirements for Address Data," page 35.	ics_auth (See description)	String (50)
bill_country	Country of the billing address. Use the two-character *ISO Standard Country Codes*. **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting. See "Relaxed Requirements for Address Data," page 35.	ics_auth (See description)	String (2)
bill_state	State or province of the billing address. For an address in the U.S. or Canada, use the *State, Province, and Territory Codes for the United States and Canada*. **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting. See "Relaxed Requirements for Address Data," page 35.	ics_auth (See description)	String (2)

1 The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to Cybersource. Visa Platform Connect creates the TC 33 Capture file at the end of the day and sends it to the merchant's acquirer, who uses this information to facilitate end-of-day clearing processing with payment card companies.

Table 5 Request Fields (Continued)

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
bill_zip	Postal code for the billing address. The postal code must consist of 5 to 9 digits. When the billing country is the U.S., the 9-digit postal code must follow this format: [5 digits][dash][4 digits] **Example** 12345-6789 When the billing country is Canada, the 6-digit postal code must follow this format: [alpha][numeric][alpha][space] [numeric][alpha][numeric] **Example** A1B 2C3 **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting. See "Relaxed Requirements for Address Data," page 35.	ics_auth (See description)	String (9)
cavv	***Visa*** Cryptogram for payment network tokens transactions. The value for this field must be 28-character base64 or 40-character hex binary. All cryptograms use one of these formats. ***American Express*** For a 20-byte cryptogram, set this field to the cryptogram for payment network tokens transactions. For a 40-byte cryptogram, set this field to block A of the cryptogram for payment network tokens transactions. The value for this field must be 28-character base64 or 40-character hex binary. All cryptograms use one of these formats.	ics_auth (R)	String (40)
currency	Currency used for the order: USD	ics_auth (R)	String (5)
customer_cc_expmo	Two-digit month in which the payment network token expires. Format: MM. Possible values: `01` through `12`.	ics_auth (R)	String (2)
customer_cc_expyr	Four-digit year in which the payment network token expires. Format: YYYY.	ics_auth (R)	Nonnegative integer (4)

1 The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to Cybersource. Visa Platform Connect creates the TC 33 Capture file at the end of the day and sends it to the merchant's acquirer, who uses this information to facilitate end-of-day clearing processing with payment card companies.

Table 5 Request Fields (Continued)

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
customer_cc_number	The payment network token value. This value is obtained by decrypting the customer's encrypted payment data.	ics_auth (R)	Nonnegative integer (20)
customer_email	Customer's email address. **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting. See "Relaxed Requirements for Address Data," page 35.	ics_auth (See description)	String (255)
customer_firstname	Customer's first name. For a credit card transaction, this name must match the name on the card. **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting. See "Relaxed Requirements for Address Data," page 35.	ics_auth (See description)	String (60)
customer_ipaddress	Customer's IP address.	ics_auth (O)	String (15)
customer_lastname	Customer's last name. For a credit card transaction, this name must match the name on the card. **Important** It is your responsibility to determine whether a field is required for the transaction you are requesting. See "Relaxed Requirements for Address Data," page 35.	ics_auth (See description)	String (60)
customer_phone	Customer's phone number. It is recommended that you include the country code when the order is from outside the U.S.	ics_auth (O)	String (15)
eci_raw	Raw electronic commerce indicator (ECI).	ics_auth	String (2)
directory_server_transaction_id	Identifier generated during the authentication transaction by the Mastercard Directory Server and passed back with the authentication results.	ics_auth (O)	String (36)

1 The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to Cybersource. Visa Platform Connect creates the TC 33 Capture file at the end of the day and sends it to the merchant's acquirer, who uses this information to facilitate end-of-day clearing processing with payment card companies.

Table 5 Request Fields (Continued)

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
e_commerce_indicator	For a payment network tokens transaction. Possible values: ■ `aesk`: American Express card type ■ `spa`: Mastercard card type ■ `internet`: Visa card type	ics_auth (O)	String (20)
encrypted_payment_data	The encrypted payment data value. If you are using the Cybersource Decryption option, populate this field with the encrypted payment data value returned from Samsung Pay in the *3ds.data* block. See "Cybersource Decryption," page 26.	ics_auth (R)	
encrypted_payment_ descriptor	Format of the encrypted payment data. The value for Samsung Pay is `RklEPUNPTU1PTi5TQU1TVU5HLklO QVBQLlBBWU1FTlQ=`	ics_auth (R)	
grand_total_amount	Grand total for the order. This value cannot be negative. You can include a decimal point (.), but you cannot include any other special characters. Cybersource truncates the amount to the correct number of decimal places.	ics_auth (R)	Decimal (15)
ics_applications	Cybersource services to process for the request: `ics_auth`	ics_auth (R)	String (255)
merchant_id	Your Cybersource merchant ID. Use the same merchant ID for evaluation, testing, and production.	ics_auth (R)	String (30)
merchant_ref_number	Merchant-generated order reference or tracking number. Cybersource recommends that you send a unique value for each transaction so that you can perform meaningful searches for the transaction. For information about tracking orders, see *Getting Started with Cybersource Advanced for the SCMP API* (PDF | HTML).	ics_auth (R)	String (50)

1 The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to Cybersource. Visa Platform Connect creates the TC 33 Capture file at the end of the day and sends it to the merchant's acquirer, who uses this information to facilitate end-of-day clearing processing with payment card companies.

Table 5 Request Fields (Continued)

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
network_token_ cryptogram	Token authentication verification value cryptogram. For token-based transactions with 3D Secure or Identity Check, you must submit both types of cryptograms: network token and 3D Secure/Identity Check. The value for this field must be 28-character Base64 or 40-character hex binary. All cryptograms use one of these formats.	ics_auth (O)	String (40)
pa_specification_version	The 3D Secure version that you used for strong customer authentication (SCA); for example, 3D Secure 1.0.2 or 2.0.0.	ics_auth (O)	String (20)
payment_network_token_ assurance_level	Confidence level of the tokenization. This value is assigned by the token service provider. **Note** This field is supported only for FDC Nashville Global.	ics_auth (O)	String (2)
payment_network_token_ device_tech_type	Type of technology used in the device to store token data. Possible value: `002`: Host card emulation (HCE) Emulation of a smart card by using software to create a virtual and exact representation of the card. Sensitive data is stored in a database that is hosted in the cloud. For storing payment credentials, a database must meet very stringent security requirements that exceed PCI DSS. **Note** This field is supported only for FDC Compass.	ics_auth (O)	Integer (3)
payment_network_token_ requestor_id	Value that identifies your business and indicates that the cardholder's account number is tokenized. This value is assigned by the token service provider and is unique within the token service provider's database. **Note** This field is supported only for FDC Nashville Global and Chase Paymentech Solutions.	ics_auth (O)	Integer (1)

1 The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to Cybersource. Visa Platform Connect creates the TC 33 Capture file at the end of the day and sends it to the merchant's acquirer, who uses this information to facilitate end-of-day clearing processing with payment card companies.

Table 5 Request Fields (Continued)

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
payment_network_token_transaction_type	Type of transaction that provided the token data. This value does not specify the token service provider; it specifies the entity that provided you with information about the token. Set the value for this field to `1`.	ics_auth (R)	String (1)
payment_solution	Identifies Samsung Pay as the payment solution that is being used for the transaction: Set the value for this field to `008`. **Note** This unique ID differentiates digital solution transactions within the Cybersource platform for reporting purposes.	ics_auth (R)	String (3)
surcharge_amount	The surcharge amount is included in the total transaction amount but is passed in a separate field to the issuer and acquirer for tracking. The issuer can provide information about the surcharge amount to the customer. This field is supported only for Visa Platform Connect.	ics_auth (O)	String (15)
ucaf_authentication_data	Cryptogram for payment network tokens transactions with Mastercard.	ics_auth (R)	String (32)
ucaf_collection_indicator	Required field for payment network tokens transactions with Mastercard. Set the value for this field to `2`.	ics_auth (R)	String with numbers only (1)
xid	***American Express*** For a 20-byte cryptogram, set this field to the cryptogram for payment network tokens transactions. For a 40-byte cryptogram, set this field to block A of the cryptogram for payment network tokens transactions. See "Merchant Decryption," page 20. The value for this field must be 28-character base64 or 40-character hex binary. All cryptograms use one of these formats.	ics_auth (R)	String (40)

1 The TC 33 Capture file contains information about the purchases and refunds that a merchant submits to Cybersource. Visa Platform Connect creates the TC 33 Capture file at the end of the day and sends it to the merchant's acquirer, who uses this information to facilitate end-of-day clearing processing with payment card companies.

Offer-Level Fields

Table 6 Offer-Level Fields

Field	Description	Used By: Required (R) or Optional (O)	Data Type (Length)
amount	Per-item price of the product. This value cannot be negative.You can include a decimal point (.), but you cannot include any other special characters.	ics_auth (See description)	Decimal (15)
merchant_product_sku	Identification code for the product. This field is required when the **product_code** value is not `default` or one of the values related to shipping and/or handling.	ics_auth (See description)	String (255)
product_code	Type of product. This value is used to determine the product category: electronic, handling, physical, service, or shipping. The default is `default`.	ics_auth (See description)	String (255)
product_name	Name of the product. This field is required when the **product_code** value is not `default` or one of the values related to shipping and/or handling.	ics_auth (See description)	String (255)
quantity	The default is 1. This field is required when the **product_code** value is not `default` or one of the values related to shipping and/or handling.	ics_auth (See description)	Integer (10)
tax_amount	Total tax to apply to the product. This value cannot be negative.	ics_auth (See description)	String (15)

Reply Fields

Because Cybersource can add reply fields, reply codes, and reply flags at any time:

- You must parse the reply data according to the names of the fields instead of the field order in the reply. For more information about parsing reply fields, see the documentation for your client.
- Your error handler should be able to process new reply codes and reply flags without problems.
- Your error handler should use the **ics_rcode** field to determine the result if it receives a reply flag that it does not recognize.

Your payment processor can include additional API reply fields that are not documented in this guide. See *Credit Card Services Using the SCMP API* (PDF | HTML) for detailed descriptions of additional API reply fields.

Table 7 Reply Fields

Field	Description	Returned By	Data Type & Length
auth_auth_amount	Amount that was authorized.	ics_auth	Decimal (15)
auth_auth_avs	AVS result code. See *Credit Card Services Using the SCMP API* (PDF | HTML) for a detailed list of AVS values.	ics_auth	String (1)
auth_auth_code	Authorization code. Returned only when the processor returns this value.	ics_auth	String (7)
auth_auth_response	For most processors, this value is the error message sent directly from the bank. Returned only when the processor returns this value.	ics_auth	String (10)
auth_auth_time	Time of authorization in UTC. See "Data Type Definitions," page 35.	ics_auth	Date and time (20)
auth_avs_raw	AVS result code sent directly from the processor. Returned only when the processor returns this value.	ics_auth	String (10)
auth_rcode	Indicates whether the service request was successful. Possible values: ■ `-1`: An error occurred. ■ `0`: The request was declined. ■ `1`: The request was successful.	ics_auth	Integer (1)

Table 7 Reply Fields (Continued)

Field	Description	Returned By	Data Type & Length
auth_rflag	One-word description of the result of the entire request. See *Credit Card Services Using the SCMP API* (PDF | HTML) for a detailed list of **rflag** values.	ics_auth	String (50)
auth_rmsg	Message that explains the reply flag **auth_rflag**. Do not display this message to the customer, and do not use this field to write an error handler.	ics_auth	String (255)
auth_trans_ref_no	Reference number for the transaction. This value is not returned for all processors.	ics_auth	String (60)
card_suffix	Last four digits of the cardholder's account number. This field is returned only for tokenized transactions. You can use this value on the receipt that you give to the cardholder. This field is returned only for FDC Nashville Global.	ics_auth	String (4)
currency	Currency used for the order. For the possible values, see the *ISO Standard Currency Codes*.	ics_auth	String (5)
directory_server_transaction_id	Identifier generated during the authentication transaction by the Mastercard Directory Server and passed back with the authentication results.	pa_enroll (O) pa_validate (O)	String (36)
ics_rcode	Indicates whether the service request was successful. Possible values: ■ `-1`: An error occurred. ■ `0`: The request was declined. ■ `1`: The request was successful.	ics_auth	Integer (1)
ics_rflag	One-word description of the result of the entire request. See *Credit Card Services Using the SCMP API* (PDF | HTML) for a detailed list of **rflag** values.	ics_auth	String (50)
ics_rmsg	Message that explains the reply flag **ics_rflag**. Do not display this message to the customer, and do not use this field to write an error handler.	ics_auth	String (255)

Table 7 Reply Fields (Continued)

Field	Description	Returned By	Data Type & Length
merchant_ref_number	Order reference or tracking number that you provided in the request. If you included multi-byte characters in this field in the request, the returned value might include corrupted characters.	ics_auth	String (50)
request_id	Identifier for the request.	ics_auth	String (26)
request_token	Request token data created by Cybersource for each reply. The field is an encoded string that contains no confidential information such as an account or card verification number. The string can contain a maximum of 256 characters.	ics_auth	String (256)
token_expiration_month	Month in which the token expires. Cybersource includes this field in the reply message when it decrypts the payment blob for the tokenized transaction. Format: MM. Possible values: `01` through `12`.	ics_auth	String (2)
token_expiration_year	Year in which the token expires. Cybersource includes this field in the reply message when it decrypts the payment blob for the tokenized transaction. Format: YYYY.	ics_auth	String (4)
token_prefix	First 6 digits of token. Cybersource includes this field in the reply message when it decrypts the payment blob for the tokenized transaction.	ics_auth	String (6)
token_suffix	Last 4 digits of token. Cybersource includes this field in the reply message when it decrypts the payment blob for the tokenized transaction.	ics_auth	String (4)