

CryptoCheckout

Plugin Setup & User Manual

Card-to-USDC Settlement for WooCommerce

Version 1.0 · For WooCommerce 7.0+ · PHP 8.0+

Overview

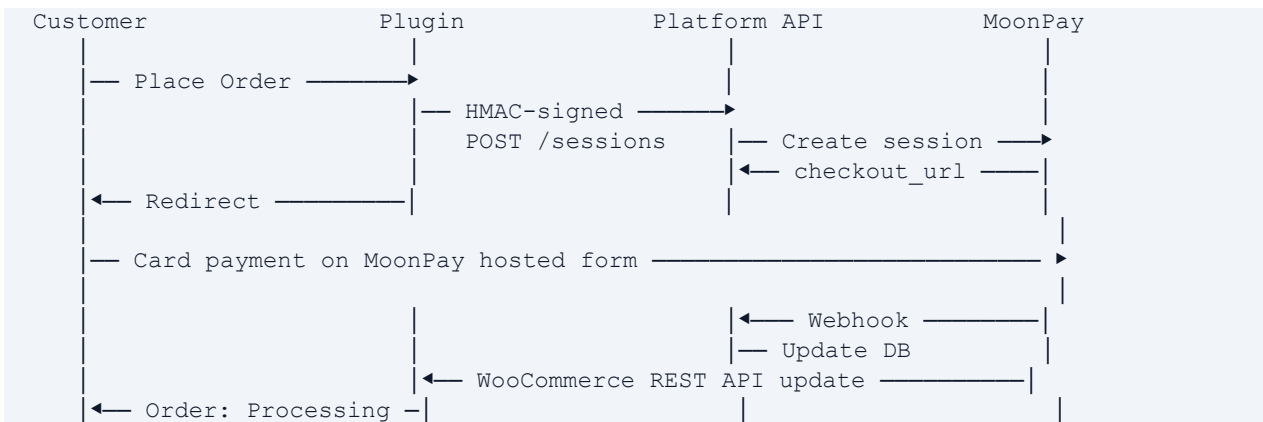
CryptoCheckout lets your WooCommerce store accept card payments settled as USDC cryptocurrency. Customers pay with a standard Visa or Mastercard through a MoonPay-hosted checkout form. You receive USDC on Polygon or Base — automatically, without ever seeing a card number.

What happens	Who handles it
Customer enters card details	MoonPay (fully PCI-compliant hosted form)
Card is charged	MoonPay payment processor
USDC is sent to your wallet	MoonPay on-chain settlement
Webhook fires with transaction details	MoonPay → CryptoCheckout platform
WooCommerce order updates to "Processing"	CryptoCheckout platform via REST API

i We never touch your card data

The CryptoCheckout plugin and platform never see, store, or process any card numbers, CVVs, or billing details. MoonPay handles all PCI compliance.

The Payment Flow



Key principle

The return URL never marks an order paid. Only a verified, idempotency-protected webhook event from MoonPay triggers a status change. This prevents double-processing and race conditions.

Before You Begin

Gather everything below before starting setup. This takes 15–20 minutes total.

System Requirements

Requirement	Minimum Version	How to Check
WordPress	6.0 or newer	Dashboard → At a Glance (bottom right)
WooCommerce	7.0 or newer	Plugins → Installed Plugins → WooCommerce
PHP	8.0 or newer	Tools → Site Health → Info → Server
SSL Certificate	Required (HTTPS)	Your store URL must start with https://

Accounts You Need

- MoonPay sandbox account — free at dashboard.moonpay.com
- Your CryptoCheckout platform deployed on Vercel (see Architecture Guide)
- Supabase project with the schema applied
- A USDC wallet address to receive settlements (MetaMask, Coinbase Wallet, etc.)

Sign up for MoonPay first

Create your free MoonPay sandbox account at dashboard.moonpay.com before continuing. The sandbox is available immediately with no KYC required.

1

Install the Plugin

Upload and activate the CryptoCheckout plugin in WordPress.

Method A — Upload ZIP (Recommended)

1. Go to your WordPress admin → Plugins → Add New Plugin.
2. Click the "Upload Plugin" button at the top of the page.
3. Click "Choose File" and select `crypto-checkout-gateway.zip`.
4. Click "Install Now".
5. Click "Activate Plugin".

Method B — Manual FTP

If you prefer to install via FTP or SSH:

```
# 1. Unzip the plugin file
unzip crypto-checkout-gateway.zip

# 2. Upload the entire folder to:
/your-wordpress-root/wp-content/plugins/crypto-checkout-gateway/

# 3. In WordPress admin:
#   Plugins → Installed Plugins → find "Crypto Checkout" → Activate
```

✓ Activation confirmed

After activation you will see "CryptoCheckout Setup Wizard" in your WordPress admin sidebar under WooCommerce, and a banner notice at the top of the screen.

2**Run the Setup Wizard**

Register your store with the CryptoCheckout platform in 3 steps.

Go to WooCommerce → CryptoCheckout Setup Wizard in your WordPress admin sidebar. The wizard registers your store and generates your HMAC secret and sandbox URL.

Wizard Step 1 of 3 — Store Info

Field	What to Enter
Store Name	A friendly label, e.g. "My Clothing Store". Used for display only.
Platform URL	Leave as https://api.cryptocheckout.io unless you are self-hosting the platform on a custom domain.

Wizard Step 2 of 3 — API Keys**Complete Step 4 first**

You need WooCommerce REST API credentials for this step. Read Step 4 of this guide first to generate your Consumer Key and Consumer Secret, then return here.

Field	What to Enter
Consumer Key	Starts with ck_ — generated in WooCommerce → Settings → Advanced → REST API.
Consumer Secret	Starts with cs_ — shown only once when you generate the key.
Store URL	Your full WooCommerce store URL including https:// , e.g. https://mystore.com .

Wizard Step 3 of 3 — Confirmation**Save your HMAC secret immediately — it is shown only once**

The 64-character HMAC secret is the shared signing key between your plugin and the platform. Copy it now and save it somewhere secure. If you lose it you must re-register your store.

After clicking "Register Store", you will receive:

Item	Example	What to do with it
Sandbox URL	myshop.cryptocheckout.io	Bookmark this — it's your private test dashboard.
HMAC Secret	64-character hex string	Paste into plugin Settings → HMAC Secret Key.
Store Slug	myshop	Paste into plugin Settings → Store Slug.

3**Get Your MoonPay API Keys**

Retrieve sandbox API keys and configure the webhook URL in MoonPay.

6. Log into dashboard.moonpay.com. Create a free account if you do not have one yet.
7. In the sidebar, go to Developers → API Keys.
8. Under the Sandbox section, copy all three keys (see table below).
9. Go to Developers → Webhooks → Add Endpoint.
10. Paste the webhook URL below and select "Transaction status updates".
11. Click Save. Copy the Webhook Signing Key that appears.

Webhook URL to paste in MoonPay:

```
https://api.cryptocheckout.io/api/v1/webhooks/moonpay
```

MoonPay Key	Starts with	Environment Variable
Publishable Key	pk_test_...	MOONPAY_API_KEY
Secret Key	sk_test_...	MOONPAY_SECRET_KEY
Webhook Signing Key	wk_test_...	MOONPAY_WEBHOOK_KEY
Widget URL	(set fixed value)	MOONPAY_WIDGET_URL = https://buy-sandbox.moonpay.com

Paste these values into your Vercel project environment variables:

```
MOONPAY_API_KEY           = pk_test_...
MOONPAY_SECRET_KEY       = sk_test_...
MOONPAY_WEBHOOK_KEY      = wk_test_...
MOONPAY_WIDGET_URL       = https://buy-sandbox.moonpay.com
SETTLEMENT_WALLET_ADDRESS = 0xYourUSDCWalletAddress
```

**Select the correct webhook event**

In MoonPay's webhook settings, make sure you select "Transaction status updates". Without this, your orders will never update automatically when payment is confirmed.

4**Create WooCommerce REST API Keys**

Grant the platform read/write access to update order status.

12. Go to WooCommerce → Settings → Advanced → REST API.
13. Click the "Add Key" button.
14. Fill in the form as shown in the table below.
15. Click "Generate API Key".
16. IMMEDIATELY copy both the Consumer Key (ck_...) and Consumer Secret (cs_...) — the secret is shown only once.
17. Paste these credentials into the Setup Wizard (Step 2 of 3).

Field	Value	Notes
Description	CryptoCheckout	Any label to identify this key
User	Your admin user	Must be an Administrator role
Permissions	Read/Write	⚠ Read-only will NOT work — orders cannot update

! Consumer Secret shown only once

After clicking "Generate API Key", WooCommerce shows the Consumer Secret one time only. Copy it immediately to a password manager or secure notes. If you miss it, delete the key and generate a new one.

5**Configure Plugin Settings**

Set up the gateway in WooCommerce → Settings → Payments.

18. Go to WooCommerce → Settings → Payments.
19. Find "Crypto Checkout (Card to USDC)" and click "Manage".
20. Fill in all settings as described in the reference table below.
21. Click "Save changes" at the bottom of the page.

Setting	Recommended Value	Notes
Enable/Disable	Checked	Must be enabled for the gateway to appear at checkout.
Title	Pay by Card (Crypto Settlement)	Customers see this label at checkout.
Platform API URL	https://api.cryptocheckout.io	Your Vercel deployment URL.
HMAC Secret Key	Paste from Setup Wizard	64-character hex key from registration. Never share this.
Store Slug	From Setup Wizard	e.g. "myshop" – your sandbox URL prefix.
Fee Percent	0 – 5%	MoonPay charges ~2.5%. You can absorb it or pass to customer.
Fee Mode	Merchant Pays or Customer Pays	Customer Pays adds a transparent fee line item at checkout.
Settlement Chain	Polygon	Lower gas fees than Ethereum. Base also supported.
Min Amount (USD)	10	Gateway hidden if order total is below this value.
Max Amount (USD)	10000	Gateway hidden if order total is above this value.
Sandbox Mode	Checked (for testing)	Uncheck ONLY when you are ready to go live.

i Tip: Customer Pays mode

When Fee Mode is set to "Customer Pays", a transparent line item appears at checkout showing the exact processing fee. This is recommended so customers understand the full cost before placing their order.

6**Test Your Setup**

Run a complete end-to-end test in sandbox mode before accepting real payments.

MoonPay Sandbox Test Cards

Use these card numbers in the MoonPay sandbox. Use any future expiry date and any 3-digit CVV.

Card Number	Result	Use For
4000 0566 5566 5556	✓ Successful payment	Primary happy-path test. Order should reach "Processing".
4000 0000 0000 0002	✗ Card declined	Tests failed payment. Order should reach "Failed".
4000 0000 0000 9995	✗ Insufficient funds	Tests a second failure scenario.

End-to-End Test Checklist

Work through each step in order and confirm each result before moving on.

Add any product to cart and go to checkout page	✓ Crypto Checkout payment option is visible
Select "Pay by Card (Crypto Settlement)"	✓ Description and fee line item (if Customer Pays mode) appear
Click "Place Order"	✓ Browser redirects to MoonPay – a card form appears
Enter test card 4000 0566 5566 5556 with any future expiry and CVV	✓ Payment processes successfully
Return to your store	✓ "Payment Processing" message shown. Order is NOT yet marked paid.
Wait 10–30 seconds for MoonPay webhook	✓ Webhook fires in the background

Check WooCommerce → Orders	✓ Order status has changed to "Processing"
Open the order in admin	✓ Order note shows transaction details; metadata fields are populated
Visit {slug}.cryptocheckout.io	✓ Session shows status "settled" and webhook event is logged
Replay the same webhook to your platform URL	✓ Returns 200, no double-update to WooCommerce

Order Metadata After Successful Test

After a successful sandbox test, open the order in WooCommerce admin and verify these custom fields are populated:

Metadata Key	Example Value
_ccg_session_id	sess_XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
_ccg_provider	moonpay
_ccg_provider_tx_id	mp_XXXXXXXXXXXXXXXXXXXX
_ccg_settlement_chain	polygon
_ccg_settlement_wallet	OxYourWalletAddress
_ccg_tx_hash	OxAbcDef1234... (on-chain hash)
_ccg_sandbox	yes

Going Live

! Do not go live until all sandbox tests pass

Complete every item in the Step 6 checklist before switching to production. Live transactions cannot be reversed from the plugin.

MoonPay

- In MoonPay dashboard, switch from Sandbox to Live API keys.
- Update your Vercel environment variables: MOONPAY_API_KEY, MOONPAY_SECRET_KEY, MOONPAY_WEBHOOK_KEY.
- Set MOONPAY_WIDGET_URL = https://buy.moonpay.com (remove -sandbox).
- Update the webhook URL in MoonPay dashboard to your production domain.
- Verify MoonPay KYC and compliance requirements for your country.

Plugin Settings

- In WooCommerce → Payments → Crypto Checkout → Manage: uncheck Sandbox Mode.
- Verify the HMAC Secret Key matches your live store record in Supabase.
- Set Min/Max amounts appropriate for your product prices.

WooCommerce

- Confirm your WooCommerce REST API key has Read/Write permissions.
- Place one low-value live test order to verify end-to-end flow.
- Verify order metadata populates correctly on the live order.

Platform & Infrastructure

- Set NODE_ENV=production in Vercel environment variables.
- Confirm wildcard domain *.cryptocheckout.io is verified in Vercel Dashboard → Domains.
- Review audit_logs table in Supabase for any unresolved errors.
- Enable Vercel log drain or alerting for production monitoring.

Troubleshooting

Solutions to the most common setup and runtime issues.

Payment method not showing at checkout

Cause	Gateway is disabled, or order total is outside the min/max range, or HMAC Secret Key or Platform API URL is missing in settings.
Fix	Check WooCommerce → Settings → Payments → Crypto Checkout is enabled (toggle on). Verify all required fields are filled. Confirm cart total is between your Min Amount and Max Amount settings.

Order stays "On Hold" and never updates to "Processing"

Cause	MoonPay webhook is not reaching the platform. The webhook URL is wrong, the wrong event type is selected, or a network issue is blocking the request.
Fix	In MoonPay Dashboard → Developers → Webhooks, confirm the endpoint URL is correct and "Transaction status updates" is selected. Check Vercel → Functions → Logs for any incoming requests. Try triggering a test webhook from MoonPay's dashboard.

"Unauthorized" error when placing an order

Cause	The HMAC Secret Key in the plugin does not match the hmac_secret stored in the Supabase stores table for your store domain.
Fix	Open the Supabase Table Editor → stores table, find your store row, and copy the exact hmac_secret value. Paste it into WooCommerce → Payments → Crypto Checkout → Manage → HMAC Secret Key. Save settings.

Sandbox dashboard at {slug}.cryptocheckout.io shows 404

Cause	The wildcard domain *.cryptocheckout.io is not configured in Vercel, or the store slug does not exist in the database.
Fix	In Vercel → your project → Settings → Domains, confirm *.cryptocheckout.io is added and shows as verified. Check that your DNS has a wildcard A record pointing to Vercel's IP. Then confirm the store_slug column in the Supabase stores table matches the slug in your URL.

Fee line item not appearing at checkout

Cause	Fee Mode is set to "Merchant Pays" instead of "Customer Pays", or Fee Percent is set to 0.
Fix	In plugin Settings, change Fee Mode to "Customer Pays" and set Fee Percent to a non-zero value (e.g. 2.5). Save settings and reload the checkout page.

WooCommerce REST API returning 401 Unauthorized

Cause	The Consumer Key or Consumer Secret stored in the platform is wrong or expired.
Fix	Go to WooCommerce → Settings → Advanced → REST API. Delete the existing key and generate a new one with Read/Write permissions. Run the Setup Wizard again or manually update the Supabase stores table row with the new credentials.

Duplicate webhook processing

Cause	This should never happen due to idempotency protection, but if you are seeing double order notes, the UNIQUE(provider, event_id) database constraint may not be applied.
Fix	Check the Supabase SQL editor and confirm the UNIQUE constraint exists on the webhook_events table. Run: <code>SELECT * FROM webhook_events WHERE event_id = '<your_event_id>'</code> and confirm only one row exists.

i First places to check

Open your sandbox at `{slug}.cryptocheckout.io` — sessions and webhook events appear in real time. Check Vercel → Functions → Logs for API errors. Check Supabase → Table Editor → audit_logs for the full history of every session.

Quick Reference

All Environment Variables

Variable	Description	Example / Default
NEXT_PUBLIC_BASE_DOMAIN	Root domain for subdomains	cryptocheckout.io
SUPABASE_URL	Supabase project API URL	https://xxx.supabase.co
SUPABASE_SERVICE_ROLE_KEY	Service role key (secret — never expose)	eyJhbGci...
SETTLEMENT_WALLET_ADDRESS	Your USDC receiving wallet	OxYour...
MOONPAY_API_KEY	Publishable API key	pk_test_ or pk_live_
MOONPAY_SECRET_KEY	Secret key for URL signing	sk_test_ or sk_live_
MOONPAY_WEBHOOK_KEY	Webhook signing key	wk_test_ or wk_live_
MOONPAY_WIDGET_URL	Widget base URL	buy-sandbox.moonpay.com or buy.moonpay.com
LOG_LEVEL	Logging verbosity	info (production), debug (dev)

WooCommerce Order Status Mapping

Payment Status	WooCommerce Status	Meaning
created / pending	On Hold	Session created, customer on MoonPay form
paid / settled	Processing	Card charged and USDC settled on-chain
failed	Failed	Card declined or payment error
expired	Failed	MoonPay session timed out
refunded	Refunded	USDC refund issued by MoonPay

Useful URLs

Resource	URL
MoonPay Dashboard	dashboard.moonpay.com

Resource	URL
MoonPay API Docs	docs.moonpay.com
Supabase Dashboard	app.supabase.com
Vercel Dashboard	vercel.com/dashboard
Your Sandbox	{slug}.cryptocheckout.io
Platform Health Check	api.cryptocheckout.io/api/health