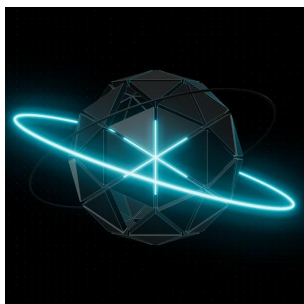




CIPHRA

AI that computes on ciphertext

An FHE coprocessor network for LLMs, launched as a protected Uniswap v4 market on Programmable · Whitepaper v0.2 · October 2026

Token: \$CPH Chain: Ethereum Category: Confidential computing X: @ciphra_agent Telegram: t.me/ciphra_agent

What changed since v0.1. The token section is no longer "to be set by a launchpad". \$CPH launches through Programmable's Custom Launch on Ethereum, inside Programmable's mandatory launch-protection hook for Uniswap v4: 100% of the supply goes into the pool and stays there, fees are fixed in the hook for ever, and every creator fee flows into one public contract, the **Cipher Vault**. Holders steer the vault with the **Blind Ballot**: a sealed vote that, from phase 2, is tallied by the CIPHRA network itself, on encrypted ballots.

This draft describes planned technology and products. Performance figures and dates are targets. It is not an offer of securities or investment advice.

Contents

1. Every prompt is a confession	3
2. Why today's options fall short	3
3. What CIPHRA is	3
4. How it works	4
5. The technology	4
5.1 Schemes	4
5.2 Making models FHE-friendly	4
5.3 Hardware acceleration	4
5.4 Performance, stated honestly	5
6. Products	5
7. Who it is for	5
8. The \$CPH token	5
8.1 Supply and allocation	6
9. Launch on Programmable	6
9.1 Fees, fixed for ever	6
9.2 Born protected	7
10. The Cipher Vault and the Blind Ballot	7
10.1 Where every creator fee goes	7
10.2 How the Blind Ballot works	8
10.3 Powers and limits	8
11. Roadmap	8
12. Security model and honest limits	9
13. Team and partners	9
14. Risks and disclaimer	9
Appendix A. Launch parameters	10

1. Every prompt is a confession

A doctor pastes a patient's scan report into an AI assistant and asks for a second opinion. A lawyer uploads a draft settlement to check the wording. A trader asks a model to stress-test a position nobody else is supposed to know about. A teenager types something into a chatbot at 2 a.m. that they have never told anyone.

Each of these is a confession. In almost every AI product today, that confession arrives at someone else's server in plain text. The model has to read it to answer it, and so does the infrastructure around the model: logs, caches, monitoring, the engineers on call.

Most of the time nothing bad happens. But "most of the time" is not good enough for a hospital, a bank or a government, which is why the organizations with the most valuable data are often the slowest to use AI. They are not afraid of the model. They are afraid of the moment their data becomes readable.

What if the model could answer without ever being able to read the question?

That is a branch of cryptography called fully homomorphic encryption (FHE), first shown to be possible in 2009. FHE lets a computer run calculations on encrypted data and return an encrypted result that only the data owner can unlock. The server does the work. It never sees the input, and it never sees the answer.

For years FHE was too slow for anything but toy problems. That is changing: better schemes, open-source libraries and dedicated hardware have cut the gap by orders of magnitude, and government and industry programs are building FHE accelerator chips. CIPHERA exists to turn that progress into something developers can use today and scale tomorrow.

2. Why today's options fall short

Teams that want private AI today choose between four imperfect options:

Option	What it protects	What it leaves open
Standard AI API with a privacy policy	Legal promises, contracts	Data is readable by the provider and its infrastructure
Self-hosting a model on-premise	Data stays inside the company	Expensive hardware, limited model quality, heavy operations
Trusted execution environments (TEE)	Data is isolated inside secure hardware	Trust in the chip vendor; a history of side-channel attacks
Anonymizing data before sending	Removes obvious identifiers	Context can re-identify people; useful detail is lost

FHE is the only approach where security does not depend on trusting the operator, the hardware vendor or a policy document. The data is protected by mathematics. Its weakness has always been speed, and that is where CIPHERA focuses.

3. What CIPHERA is

CIPHERA is an FHE coprocessor network for language models. Developers send encrypted prompts and receive encrypted outputs. Independent node operators run the encrypted computation on accelerated hardware and are paid per job. The encryption keys never leave the user.

- **For developers:** an SDK and API that look like a normal AI inference API, with encryption and decryption handled on the client.
- **For node operators:** software that turns GPU servers, and later dedicated FHE accelerators, into paid coprocessors.
- **For end users:** AI features where "we can't see your data" is a technical fact, not a promise.
- **For \$CPH holders:** a public vault, funded by every trade, whose direction they decide by sealed ballot (section 10).

4. How it works

1. **Key generation.** The client SDK creates a secret key on the user's device and a public evaluation key that lets servers compute on ciphertext without decrypting it.
2. **Encryption.** The prompt is tokenized and embedded locally, then encrypted.
3. **Encrypted inference.** A CIPHRA node runs the model's layers on the ciphertext. It sees only encrypted numbers in and encrypted numbers out.
4. **Verification.** The client checks that the agreed model was run. At launch this uses redundant execution on independent nodes with matching results; succinct proofs for FHE computation will be added as that research matures.
5. **Decryption.** Only the client's secret key can turn the result into readable text.

At no step does the network, the node operator or CIPHRA itself hold a key that can read the user's data.

5. The technology

5.1 Schemes

CIPHRA builds on established, peer-reviewed FHE schemes rather than inventing new cryptography. CKKS supports approximate arithmetic on real numbers and suits the matrix multiplications that dominate neural networks. TFHE evaluates non-linear functions quickly through programmable bootstrapping. The CIPHRA compiler chooses the scheme per layer, building on mature open-source libraries such as OpenFHE and TFHE-rs; every CIPHRA-specific component will be open-sourced and audited.

5.2 Making models FHE-friendly

Neural networks contain operations that are cheap in plain text but expensive under encryption, especially softmax, GELU and normalization. CIPHRA uses three techniques:

- Polynomial approximation of non-linear functions, tuned per model so accuracy loss stays within a stated tolerance.
- Hybrid splitting: the client runs the cheap first and last steps (tokenization, embedding, decoding) locally; the network runs the heavy middle layers encrypted.
- FHE-aware fine-tuning of open-weight models so they keep their quality after approximation.

5.3 Hardware acceleration

Stage	Hardware	Purpose
Now	Data-center GPUs	Custom CUDA kernels for NTT and key switching, on hardware operators already own

Stage	Hardware	Purpose
Next	FPGA cards	Lower latency and better energy use for production workloads
Later	FHE ASICs	Dedicated accelerator chips, sourced through hardware partnerships as they reach the market

5.4 Performance, stated honestly

Encrypted inference is still much slower and more expensive than plain inference, and will remain so for some time. CIPHRA does not claim to run frontier-size models fully encrypted at chat speed. The first targets are workloads where encrypted inference is already practical: classification, embeddings for private search, document extraction and compact language models. CIPHRA will publish latency, cost per request and accuracy against the unencrypted baseline for every supported model.

6. Products

Product	Description
CIPHRA SDK	Client libraries (Python, TypeScript) for keys, encryption, decryption and proof checking
Private Inference API	A familiar API for encrypted calls to supported models, priced per request
Private RAG	Encrypted embeddings and similarity search over private documents
CIPHRA Node	Software for operators to run encrypted workloads on GPU servers and earn fees
Model Registry	Supported models with their FHE parameters, accuracy reports and benchmarks
Blind Tally	Encrypted vote counting as a service, first used by CIPHRA's own Blind Ballot (section 10)

7. Who it is for

- **Healthcare:** triage, coding and summarization of patient records without exposing them to the AI provider.
- **Legal:** contract review and e-discovery on privileged documents.
- **Finance:** fraud scoring and risk analysis on data that cannot leave the bank in readable form.
- **Enterprise copilots:** internal assistants that read sensitive documents without the vendor seeing them.
- **On-chain agents and DAOs:** agents that act on private inputs, and organizations that want secret ballots with public, verifiable results.

8. The \$CPH token

\$CPH is the utility token of the CIPHRA network. It is designed to be used, not just held.

Use	How it works	When
Steer the Cipher Vault	Stake \$CPH and vote in the weekly Blind Ballot on where half of all creator fees go	From launch

Use	How it works	When
Pay for compute	Encrypted inference jobs are priced in \$CPH; enterprises may pay in stablecoins, converted at settlement	Phase 2
Node staking	Operators stake \$CPH to receive jobs; results that fail verification lose part of the stake	Testnet, then mainnet
Priority capacity	Stakers reserve guaranteed capacity for production workloads	Phase 3
Governance	Supported models, network fee parameters and grants	Phase 4

8.1 Supply and allocation

Allocation	Share	\$CPH	Notes
Uniswap v4 pool (ETH/\$CPH)	100%	1,000,000,000	Placed in the launch transaction as one position, locked forever
Team, investors, presale, airdrop	0%	0	None exist. The team buys from the same pool as everyone else.

The supply is fixed: no mint, no owner, no tax inside the token. The only \$CPH outside the pool at birth is the launch wallet's single opening buy, made inside the launch transaction at the pool price and capped at 2% of supply by the launch-protection rules. The team is paid the way the network is: from the treasury share of fees (section 10), never from a token reserve.

9. Launch on Programmable

CIPHRA launches as a **Programmable Custom Launch on Ethereum**. Programmable requires every new native-ETH launch to use its sealed launch-protection package (v1). CIPHRA uses that package unchanged and adds one contract of its own, the Cipher Vault, which is the hook's fixed creator-fee recipient. One Router transaction deploys everything, seeds and locks the liquidity, makes the opening buy and stamps the launch on Programmable's Explore.

Target	Contract	Role
Initializer	ProtectedNativeInitializerV1 (Programmable)	Checks the opening valuation on the Chainlink ETH/USD feed, places 100% of supply as one locked position, makes the capped opening buy
Token	ProtectedLaunchTokenV1 (Programmable)	\$CPH, 1,000,000,000 fixed supply, 18 decimals; 2% receive cap per wallet for the first 300 seconds
Hook	ProtectedNative30HookV1 (Programmable)	Takes the platform and creator fees on the ETH leg of every swap; enforces the opening protection; creates its fee vault, protection module and opening-tax vault
Cipher Vault	CipherVault (CIPHRA)	Receives every creator fee; splits it each epoch by the Blind Ballot result

9.1 Fees, fixed for ever

Fee	Rate	Recipient
-----	------	-----------

Fee	Rate	Recipient
Programmable platform fee	0.30% of the ETH leg, every buy and sell, rounded up per trade	Programmable treasury
CIPHRA creator fee	2% on buys and 2% on sells, fixed at deployment	Cipher Vault
LP fee	0%	None
All-in, after the first 30 seconds	2.3% on every trade, for ever	Nobody, CIPHRA included, can change it

9.2 Born protected

Rule	Value
Opening valuation	At least USD 15,000 fully diluted, checked at launch on the Chainlink ETH/USD feed
Opening buy fee	Starts at 99% and its extra part falls linearly to zero over 30 seconds; the extra part goes to Programmable. Only the launch wallet is exempt from the extra part.
Wallet limits	For 300 seconds every wallet, the launch wallet included, can buy at most 2% and receive at most 2% of supply. Selling never refills the budget.
After 300 seconds	Nothing limits transfers. Sells always pay the normal fee.

These rules limit early accumulation per wallet. They cannot tell whether one person controls several wallets.

10. The Cipher Vault and the Blind Ballot

The idea. A privacy network should not ask its holders to vote in public. In the Blind Ballot nobody, not even the CIPHRA team, sees how anyone voted; only the result is published. From phase 2 the ballots are counted by CIPHRA's own FHE nodes on ciphertext, so the first real customer of the network is its own treasury.

10.1 Where every creator fee goes

Anyone can call the hook fee vault's `claimCreator()` to push accrued creator fees (ETH) into the Cipher Vault, and anyone can call `distribute()`. Every distribution splits the vault's ETH in fixed shares:

Share	Destination	Purpose
50%	The Sealed Pot	Goes to the route chosen by the current epoch's Blind Ballot (below)
30%	R&D treasury wallet	Cryptography, audits, GPU time and the team
20%	Break-the-Cipher fund	Public bounties: bugs in the SDK or node, broken parameters, accuracy regressions. Security as a paid public sport.

The Sealed Pot has three routes:

Route	Where the Sealed Pot goes	Why holders might choose it
Burn	Buyback wallet: \$CPH is bought back on the pool and burned; every burn is announced with its transaction	Supply shrinks with trading activity

Route	Where the Sealed Pot goes	Why holders might choose it
Compute	Compute Fund: prepaid encrypted-inference credits and grants for developers building on CIPHERA	Usage grows the network
Nodes	Node Bootstrap: rewards paid to node operators per verified job on testnet and mainnet	More and faster coprocessors

If a ballot fails to reach quorum, the previous route stays. The default route at launch is **Burn**. Before a new route takes effect the vault distributes everything that arrived under the old one, so a vote never re-routes fees earned before it.

10.2 How the Blind Ballot works

1. **Stake.** Holders stake \$CPH in the CipherStake contract. Unstaking has a 7-day cooldown, and only stake that existed when an epoch opened can vote in it, so tokens cannot be borrowed for a single vote.
2. **Seal (days 1 to 5 of each 7-day epoch).** A voter submits a sealed ballot: in phase 1 a commitment `hash(route, salt, voter)`; from phase 2 the route encrypted to the network's threshold public key.
3. **Open (days 6 and 7).** Phase 1: voters reveal route and salt; unrevealed ballots do not count. Phase 2: CIPHERA nodes add the encrypted ballots, weighted by stake, without decrypting any of them; a threshold of nodes decrypts only the three totals.
4. **Apply.** At 00:00 UTC on Sunday the winning route is written to the Cipher Vault. Results carry the tally transcript hash, and in phase 2 the redundant-execution attestation of the nodes that computed it.

Phase 1 already hides votes until the reveal; phase 2 hides them for ever. Moving from one to the other needs no change to the token, the pool or the vault split, only the ballot contract.

10.3 Powers and limits

Actor	Can	Cannot
Anyone	Claim creator fees into the vault; call <code>distribute()</code> ; apply a finished ballot	Choose where funds go
Stakers	Vote in the Blind Ballot	Move vault funds directly
Guardian (one ordinary wallet at launch, not a multisig)	Change the recipient wallets, the ballot or itself after a 48-hour public timelock; pause the ballot, which freezes the route at Burn (safe mode); pay Break-the-Cipher bounties from that fund	Change any fee, touch the pool or its liquidity, mint, pause trading
CIPHERA team	Spend the R&D treasury share, publicly reported each quarter	Spend the Sealed Pot outside the winning route

No contract spends vault ETH at a market price by itself: buybacks, burns and grants from the recipient wallet are executed by its holder and announced, with the vault's **Distributed** events as the public ledger to check them against.

11. Roadmap

Milestones are targets and depend on funding, audits and hardware availability.

Phase	Timing	Deliverables
0 · Birth	Launch	Protected Custom Launch on Programmable; Cipher Vault and CipherStake live; first commit-reveal Blind Ballot; Break-the-Cipher fund opens
1 · Proof	Months 0 to 6	Open-source compiler alpha; encrypted classification and embedding models on GPU; public benchmarks; developer testnet with Node Bootstrap rewards
2 · Private RAG	Months 6 to 12	Private RAG and Inference API beta; first external node operators; SDK and node audit; the first Blind Ballot tallied on ciphertext by CIPHERA nodes
3 · Language models	Months 12 to 24	Encrypted inference for compact language models with hybrid splitting; FPGA nodes; enterprise pilots in healthcare and finance
4 · Scale	Month 24+	Larger models as accelerators mature; ASIC partnerships; network governance by stakers

12. Security model and honest limits

- **What is protected:** prompt content, model outputs and retrieved documents, against node operators, CIPHERA and network observers.
- **What is not hidden:** that a request was made, its size and timing, and which model was called.
- **Model weights:** in the initial design weights are known to nodes; encrypted inputs are protected, not the model.
- **Key safety:** if a user's device or secret key is compromised, encryption cannot protect their data.
- **Accuracy:** approximated models may differ slightly from the original; each supported model ships with an accuracy report.
- **Blind Ballot:** in phase 1 a ballot is hidden only until its reveal; stake amounts are public on-chain, so very large stakers can still be inferred from totals.
- **Launch contracts:** the token, hook and initializer are Programmable's launch-protection package; the Cipher Vault and CipherStake are CIPHERA's and are unaudited until the audit milestone.

13. Team and partners

To be completed before publication: core team and advisors with verifiable backgrounds in cryptography and systems, the legal entity, audit firms, hardware partners. Investors and users judge confidential-computing projects heavily on the team's credibility, so this section should not be published empty.

14. Risks and disclaimer

- **Technical:** performance targets may take longer than planned; cryptographic implementations may contain bugs despite audits.
- **Hardware:** availability and cost of FPGA and ASIC accelerators are outside the project's control.
- **Adoption:** the network depends on developers and enterprises choosing encrypted inference over cheaper alternatives.

- **Platform:** the launch depends on Programmable's review, router and indexing. A launch stamp proves origin, not safety.
- **Trading:** buys in the first 30 seconds pay up to 99% in fees; the hook uses Uniswap v4 return deltas, so some third-party interfaces may not route it.
- **Fees and treasury:** vault income depends entirely on trading volume and is never guaranteed. Bounties and grants are discretionary.
- **Regulatory and market:** rules on digital assets differ by country and change over time; the price of \$CPH may be volatile and holders may lose part or all of its value.

This document describes planned technology. It is not an offer to sell or a solicitation to buy any security, and nothing in it is investment, legal or tax advice. All figures and timelines are subject to change.

Appendix A. Launch parameters

Parameter	Value
Chain	Ethereum mainnet (chain id 1)
Launch route	Programmable Custom Launch API, profile 3.6.0, launch-protection v1, CLI 4.1.7
Pool	Uniswap v4, native ETH / \$CPH, LP fee 0, tick spacing 60
Supply	1,000,000,000 \$CPH, 18 decimals, fixed
Creator fee	200 bps buy / 200 bps sell, fixed in the hook, recipient: Cipher Vault
Platform fee	30 bps, Programmable treasury 0xD88539d3c4C460136a733A3Fd60cf6BF269079da
Opening protection	99% → normal over 30 s; 2% buyer and receiver caps for 300 s; ≥ USD 15,000 opening valuation
Vault split	50% Sealed Pot (Blind Ballot) · 30% R&D treasury · 20% Break-the-Cipher fund
Ballot epoch	7 days: seal days 1 to 5, open days 6 and 7, applied Sunday 00:00 UTC
Staking cooldown	7 days
Launch wallet and vault roles	0x8B4b6d4AB1Ad29E22cA5Fb850E8959a7439D944F: one ordinary wallet (not a multisig) is the launch wallet, the guardian and every vault recipient at launch
Guardian changes	48-hour public timelock on recipient, ballot and guardian changes; the guardian can be moved to a multisig through it
Opening buy	0.05 ETH by the launch wallet inside the launch transaction, about 6.9 million \$CPH (at most 2% of supply)
Opening valuation	About USD 17,058 at tick 187740; the launch reverts if ETH/USD is below about 2,134 at launch