Zama Confidential Blockchain Protocol Litepaper

This litepaper describes Zama's Confidential Blockchain Protocol, which enables confidential smart contracts on existing public blockchains. It includes details about the protocol and token, as well as documentation for node operators.

To read Zama's FHEVM technical white paper, please see on Github ↗.
To read the Zama Token White Paper, see this page ↗.

The blockchain confidentiality dilemma

Why do we need blockchain? This question often comes along when discussing building decentralized applications (dapps). After all, most things we use today are not blockchain based and work just fine. However, there are some applications where the cost of blindly trusting a third party and being wrong is too high, such as when dealing with financial assets, identity or governance. In those cases, consumers and companies want strong guarantees that whatever service is being provided is done correctly, while service providers want to ensure their users have the right to use the assets/data they claim.

Blockchains solve this by enabling anyone to publicly verify that a request was executed according to a predetermined logic, and that the resulting state of the overall system is correct. Service providers and their customers no longer have to trust each other, as the integrity of the transaction is guaranteed by the blockchain itself.

Archived copy · cryp2nova.com

One major issue with public verifiability however is that it requires disclosing all the transactions and data to everyone, as keeping them private would prevent verifiability in the first place. This lack of confidentiality has been a major hindrance to global adoption of blockchains, as the very data it is supposed to be used for (money, identity, …) is highly sensitive by nature. Without confidentiality, blockchain cannot reach mass adoption.

The Zama Confidential Blockchain Protocol

The Zama Confidential Blockchain Protocol (or simply the Zama Protocol) enables issuing, managing and trading assets confidentially on existing public blockchains. It is the most advanced confidentiality protocol to date, offering:

- **End-to-end encryption** of transaction inputs and state: no-one can see the data, not even node operators.
- **Composability** between confidential contracts, as well as with non-confidential ones. Developers can build on top of other contracts, tokens and dapps.
- **Programmable confidentiality**: smart contracts define who can decrypt what, meaning developers have full control over confidentiality rules in their applications.

The Zama Protocol is not a new L1 or L2, but rather a cross-chain confidentiality layer sitting on top of existing chains. As such, users don't need to bridge to a new chain and can interact with confidential dapps from wherever they choose.

It leverages Zama's state-of-the-art Fully Homomorphic Encryption (FHE) technology, which enables computing directly on encrypted data. FHE has long been considered the "holy grail" of cryptography, as it allows end-to-end encryption for any application, onchain or offchain. We believe that just like the internet went from zero encryption with HTTP to encrypting data in transit with HTTPS, the next natural step will be to use FHE to enable end-to-end encryption by default in every application, something we call HTTPZ↗.

Archived copy · cryp2nova.com

Until recently however, FHE was too slow, too limited in terms of applications it could support, and too difficult to use for developers. This is what our team at Zama has spent the last 5 years solving. We now have a highly efficient FHE technology that can support any type of application, using common programming languages such as Solidity and Python, while being over 100x faster than 5 years ago. Importantly, Zama's FHE technology is already post-quantum, meaning there is no known quantum algorithms that can break it.

While FHE is the core technology used in the Zama Protocol, we also leverage Multi-Party Computation (MPC) and Zero-Knowledge Proofs (ZK) to address the shortcomings of other confidentiality solutions:

- FHE enables confidentiality while being fully publicly verifiable (anyone can recompute the FHE operations and verify them). Using GPUs will soon allow scaling to 100+ transactions/s while dedicated hardware accelerators (FPGAs and ASICs) will enable scaling to thousands of transactions per second.
- MPC enables decentralizing the global network key, ensuring no single party can access it. Using MPC only to generate keys and decrypt data for users minimizes latency and communication, thereby making it far more scalable and decentralized than using it for private computation.
- ZK ensures the encrypted inputs provided by users were actually encrypted correctly. Using ZK only for this specific purpose makes the ZK proofs lightweight and cheap to generate in a browser or mobile app.

The table below summarizes the advantages of the Zama Protocol versus other technologies used in confidential blockchain protocols:

Archived copy · cryp2nova.com

	Zama	Other FHE	MPC	ZK
Secure	✅	✅	✅	✅
Decentralized	✅	✅	✅	✅
Verifiable	✅	✅	❌	✅
Composable	✅	✅	✅	❌
Scalable	✅	❌	✅	✅
Easy to use	✅	❌	❌	❌

Roadmap

The Zama Protocol leverages years of research and development work done at Zama. The testnet is already live, with the mainnet on Ethereum launching by end of 2025. The timeline is as follows:

- **Ethereum Mainnet (live)**. This will be the first official mainnet bringing confidentiality to Ethereum.
- **Other EVM chains (H1 2026)**. We will add more EVM chains to the Zama protocol to enable cross-chain confidential assets and applications.
- **Solana support (H2 2026)**. After an initial phase of EVM-only support, we will deploy the Zama Protocol on Solana, enabling confidential SVM applications.

Use cases

Confidential smart contracts enable a new design space for blockchain applications, in particular when applied to finance, identity and governance. If we look at web2, it is clear that most applications do not share all the data publicly, and thus it is likely that the vast majority of blockchain applications are yet to be built, now that confidentiality is no longer an issue.

Here are some example use cases:

Archived copy · cryp2nova.com

Finance

- **Confidential payments.** Stablecoins are one the most successful use case for blockchain, with trillions in yearly volume. Everything from credit card payments to salaries, remittances and banking rails is now moving onchain. One of the absolute key requirement however is confidentiality and compliance. Thanks to FHE and the Zama Protocol, this is now possible: balances and transfer amounts are kept encrypted end-to-end, while payment providers can embed compliance features into the token contract directly. You can read more about confidential, compliant payments here↗.
- **Tokenization & RWAs**. The tokenization of financial assets is one of the main adoption drivers of blockchain for large institutions. From fund shares to stocks, bonds or derivatives, there is up to $100T of assets that could potentially move onchain. Due to confidentiality and compliance issues however, TradFi institutions have had to rely on private blockchains, making it difficult to ensure interoperability between institutions. With the Zama Protocol, they can now use existing public blockchain such as Ethereum or Solana to tokenize and trade their assets, while keeping their activity and investor identity confidential. They can also ensure KYC/AML checks are done in the smart contracts directly, without revealing sensitive information to others. You can read more about this use case in the report published by JP Morgan - Kynexis↗, in which they built a proof-of-concept using Zama's technology.
- **Confidential DeFi.** DeFi has redefined finance by allowing anyone to participate and earn yield, but it suffers from two major issues: people don't like sharing how much they own, and bots front-running transactions makes it expensive for end users to swap assets onchain. FHE can solve both issues by enabling end-to-end encrypted swaps, where the amount and possibly asset is kept private at all times. Some other use cases includes confidential lending, onchain credit scoring, option pricing and more.

Archived copy · cryp2nova.com

Tokens

- **Sealed-bid auctions.** Sell assets such as NFTs or tokens in an onchain sealed-bid auction. Each participant places an encrypted bid onchain. When the auction ends, the highest bidder(s) win the item(s), without revealing any of the bids. Not only does this enable better price discovery, it also prevents bots from stealing the auction by monitoring the mempool. This is a particularly effective method for public token sales.
- **Confidential distributions.** Distributing tokens currently requires disclosing publicly how much each address receives. Whether it's for airdrops, grants, investors or developers, keeping the distributed amounts private is paramount to privacy and security onchain. With FHE, protocols can distribute their token confidentially, run vesting on those encrypted tokens, enable confidential staking and more.

Identity and Governance

- **Composable onchain identity.** Offchain, we use our identities all the time, from buying products online to booking plane tickets. Doing so onchain however would leak sensitive information such as your name, address, social security number and more. With FHE however, you can have a complete Decentralized ID (DID) + Verifiable Credentials (VC) system onchain, where your identity is encrypted while being fully composable with decentralized applications. Just like you can have account abstraction, you can now have identity abstraction. This is also essential for compliance in onchain payments and tokenization, as it can be used by smart contracts to verify claims in a decentralized, private manner.
- **Confidential governance.** The idea of onchain voting, whether for DAOs, companies or governments, has been explored for as long as blockchains exist. But having the votes cast publicly onchain can lead to biases, blackmailing, or bribing. With FHE, votes (and numbers of tokens staked) can be kept private, ensuring only the final tally is revealed, and not the individual votes.

Archived copy · cryp2nova.com

Other examples

- **Onchain corporations.** Managing a company onchain would be impossible without the promise of confidentiality. Indeed, information such as the cap table, financials, board votes, customers, and employee registers should not be disclosed publicly. With FHE, all this information could be kept onchain, allowing smart contracts to automate many day-to-day company operations.
- **Prediction markets.** Prediction markets are based on the wisdom of the crowd concept: the average prediction of a large number of people tend to be close to the correct outcome. However, this only works if participants are not biased by previous predictions. The Zama Protocol solves this by enabling prediction markets where predictions are encrypted until revealed periodically, leading to better precision in outcomes.
- **Data marketplaces for AI.** AI strives on data. With FHE, users can selectively share and sell their data with companies wishing to train AI models. More than this, models can potentially be trained encrypted, with only the result being decrypted, ensuring that users have a constant stream of revenue for their data vs selling it only once and it being used forever.

These are just some examples of what can be done today. We believe that FHE, through Zama's Protocol, will enable unprecedented liquidity, enabling users and companies to move onchain. With time and scale, it would even become possible to run entire companies, cities or even countries onchain, including their financial and identity infrastructure, elections, currency, taxes, land, car and company registries. Confidential blockchains don't just enable programmable money: they enable programmable public infrastructure.

Creating confidential applications

Creating a confidential dapp using existing solutions often requires learning a new (niche) programming language, using dedicated (and often limited) developer tools, and mastering advanced cryptographic concepts.

Archived copy · cryp2nova.com

The Zama Protocol on the other hand enables developers to create confidential dapps directly in Solidity, without any knowledge of cryptography. Developers simply need to import our library (called FHEVM) and write their logic using the provided operators. You can get started today for free by checking out the developer documentation here ↗.

The following example shows an example confidential token contract, which can be deployed on any supported chain such as Ethereum.

CRYP2NOVA

Archived copy · cryp2nova.com

```
pragma solidity ^0.8.26;

import "fhevm/lib/FHE.sol";
import { IConfidentialFungibleToken } from
"./IConfidentialFungibleToken.sol";

abstract contract ConfidentialFungibleToken is
IConfidentialFungibleToken {

    uint64 internal _totalSupply;
    string internal _name;
    string internal _symbol;

    // Balances are encrypted
    mapping(address account => euint64 balance) internal
_balances;

    // Transfer an encrypted amount
     function transfer(address to, externalEuint64
encryptedAmount, bytes calldata inputProof) public virtual
returns (euint64) {

        // Verify the input is correct and cast to euint64
        euint64 amount = FHE.fromExternal(encryptedAmount,
inputProof);

        // Check if the user has enough balance, otherwise set
the transfer amount to zero
        euint64 transferValue = FHE.select(FHE.le(amount,
_balances[msg.sender]), amount, FHE.asEuint64(0));

        // Make the transfer
        _balances[to] = FHE.add(_balances[to], transferValue);
        _balances[from] = FHE.sub(_balances[from],
transferValue);

        // Allow users to see their balances, and the contract
to update it
        FHE.allow(_balances[to], to);
        FHE.allow(_balances[from], from);
        FHE.allowThis(_balances[to]);
        FHE.allowThis(_balances[from]);

        return transferValue;
    }
}
```


Archived copy · cryp2nova.com

Simply replace the integer operations by their FHE equivalent, then specify who can decrypt the balances. Of course, developers can build much more complicated applications, such as AMMs, lending, and more. On top of the smart contract library, we also provide a Javascript SDK that streamlines the encryption and decryption client-side, making it almost invisible to end-users.

The access control system used by the Zama Protocol is extremely powerful. By allowing contracts to define who can decrypt which value in it, it makes confidentiality (and compliance) fully programmable. There is no assumption at the protocol or user level, everything is encoded in the application logic itself, allowing companies to choose whether they want to offer end-to-end encryption (aka nobody sees anything, not even the companies building the dapp), or onchain encryption (aka the web2 model: only the user and service provider see the data, but nobody else onchain).

The FHEVM library used by the Zama Protocol supports the following encrypted types and operations:

Type	Symbol	Logical	Arithmetic	Comparison	Shifts
Integer (unsigned)	euint8…256	and, or, xor, not	add, sub, mul, div, rem, neg, abs, sign	eq, neq, gt, lt, ge, le, min, max	shl, shr, rotl, rotr
Integer (signed)	eint8…256	and, or, xor, not	add, sub, mul, div, rem, neg, abs, sign	eq, neq, gt, lt, ge, le, min, max	shl, shr, rotl, rotr
Boolean	ebool	and, or, xor, not		eq, neq	
Bytes	ebytes1…256	and, or, xor, not		eq, neq	shl, shr, rotl, rotr
Address	eaddress			eq, neq	

To make deploying dapps easier, we are also building a “Zama Standard Library”: a set of audited, highly optimized smart contracts for common use cases such as:

Archived copy · cryp2nova.com

- confidential tokens and RWAs
- confidential NFTs
- wrappers to bridge between confidential assets and traditional ones
- a confidential identity stack that enables DID/VC onchain
- UniV2-style confidential AMMs
- Confidential vesting
- Confidential airdrops
- Sealed-bid auctions
- Confidential governance

We will continue adding more over time as we see new use cases appearing.

Technical details

Blockchains typically only support limited computations, making it impossible to run FHE natively on Ethereum and other L1/L2s. To address this issue, we designed the Zama Protocol based on two core ideas: symbolic execution and threshold decryption.

Symbolic execution

The idea behind symbolic execution is that whenever a contract calls the Zama FHEVM Solidity library on the host chain (the L1/L2 where the confidential dapp is deployed) to perform an FHE operation, the host chain itself doesn't do any actual FHE computation; instead, it produces a pointer to the result and emits an event to notify a network of coprocessors, who do the actual FHE computation. This has many advantages:

- The host chain does not need to change anything, run expensive FHE operations or use specific hardware.
- The host chain is not slowed down by FHE, so non-FHE transactions can be executed as fast as they always have been

Archived copy · cryp2nova.com

- FHE operations can be executed in parallel, rather than sequentially, dramatically increasing throughput.

Since all ciphertexts on the host chain are simply pointers (the actual data is stored by coprocessors), FHE operations can be chained just like regular operations, without needing to wait for the previous ones to complete. The only time we need to wait for a ciphertext to be computed is when it has to be decrypted.

From a security perspective, everything the coprocessors do is publicly verifiable, and anyone can just recompute the ciphertexts to verify the result. Initially, we use multiple coprocessors with a majority consensus, but longer term the goal is to enable anyone to compete to execute FHE operations, leveraging ZK-FHE to prove the correctness.

Threshold decryption

To maintain composability onchain, all ciphertexts need to be encrypted under the same public key. This means the private decryption key has to be secured in a way that prevents illegitimate decryption of ciphertexts. The Zama Protocol solves this by splitting the private key amongst multiple parties, using a dedicated threshold MPC protocol as its Key Management Service (KMS).

In order for a user or contract to decrypt a value, they need to first have been explicitly allowed to do so by the contract that produced the value on the host chain. Decrypting the result is then a simple request to the Zama Gateway, which acts as an orchestrator for the protocol and forwards the request to the KMS parties.

This again ensures that all decryption requests are publicly visible, and thus anyone can verify they match the access control logic defined by smart contracts.

Components

The Zama Protocol is composed of several core components:

Archived copy · cryp2nova.com

- **Host Chains**: the L1s and L2s that are supported by the Zama Protocol, and on which developers deploy their confidential dapps.
- **FHEVM Library**: the library that developers use to create confidential smart contracts.
- **FHEVM Executor**: the contract that is called by dapps to execute FHE operations on the Host Chain. Each time a contract uses an FHE operation, the Executor automatically emits an event to notify Coprocessors to compute it.
- **Access Control List (ACL)**: a smart contract deployed on each Host Chain, which keeps tracks of who can decrypt what. The ACL is central to the operations of the Zama Protocol and is used both to verify a contract is allowed to compute on an encrypted value, and that an address is allowed to decrypt it. Each time a contract allows an address to use a ciphertext, an event is emitted and relayed by Coprocessors to the Gateway, enabling the aggregation of all Host Chain ACLs into a single Gateway ACL used by the KMS to authenticate decryption requests.
- **$ZAMA token**: the native token of the Zama Protocol, used for payment of the fees and staking.
- **Gateway**: a set of smart contracts used to orchestrate the Zama Protocol, and allow users to request verification of their encrypted inputs, decryption of ciphertexts and bridging of encrypted assets between Host Chains. Each of these operations is a transaction to the Gateway contracts, and requires paying a small fee in $ZAMA tokens. While the Gateway contracts could be deployed on any L1 or L2, we opted to run a dedicated Arbitrum rollup for the Zama Protocol, ensuring maximal performance and cost efficiency. Note that our rollup serves only the Zama Protocol and doesn't allow third party contracts to be deployed on it.
- **Coprocessors**: a set of nodes responsible for 1. verifying encrypted inputs from users, 2. running the actual FHE computations and storing the resulting ciphertexts, 3. relaying ACL events to the Gateway. The Zama Protocol uses multiple coprocessors, which each commit their results to the Gateway, which in turns runs a majority consensus. All tasks performed by the coprocessors are publicly verifiable. Coprocessors can be vertically and horizontally scaled based on throughput requirements of the various confidential dapps.

Archived copy · cryp2nova.com

- **Key Management Service (KMS)**: a set of nodes running various Multi-Party Computation (MPC) protocols for key generation, CRS generation and threshold decryption. The KMS ensures that no single party can ever access the decryption keys. KMS nodes are orchestrated by the Gateway, ensuring all operations are publicly visible. Furthermore, all KMS nodes must run the MPC software inside AWS Nitro Enclaves, making it harder for operators to leak their shares while providing some level of integrity on the MPC computation. Eventually, our goal will be to use ZK-MPC to enable verifiability without hardware assumptions.
- **Operators**: a set of entities that run the Zama Protocol nodes. This includes Coprocessors and KMS nodes.

The following diagram shows the lifecycle of a confidential token transfer across the various components.

CRYP2NOVA

Archived copy · cryp2nova.com

Archived copy · cryp2nova.com

Performance

The Zama Protocol is designed to be horizontally scalable, leveraging our cutting-edge TFHE-rs ↗ library. Contrary to the sequential behavior of the EVM, the Zama Protocol parallelizes the computation of FHE operations. As long as a specific ciphertext isn't used in a sequential chain of FHE operations, Coprocessors will be able to increase the throughput simply by adding more servers.

Since we started working on the Zama Protocol, we have been able to increase throughput exponentially from 0.2 transactions per second to over 20 transactions per second on CPU, enough to make all of Ethereum encrypted.

By the end of 2026, we will migrate to GPUs, with an expected 500-1000 TPS per chain, enough to cover all L2s and most Solana use cases.

Finally, we are working on a dedicated hardware accelerator (ASIC) for FHE, which will enable 100,000+ tps / chain on a single server, enough to bring global payments confidentially onchain.

The important point here is that FHE is no longer limited by underlying algorithms, and is now mostly driven by Moore's law: the better the hardware, the better the throughput of the Zama Protocol.

CRYP2NOVA

Archived copy · cryp2nova.com

Security

The Zama Protocol uses a defense-in-depth approach, combining multiple techniques to ensure maximum security:

- We use 128 bits of security and a p-fail of 2^-128 for all FHE operations. This is far more than any other FHE scheme used in blockchain currently. Furthermore, our FHE scheme is post-quantum, meaning it is secure even against quantum computers.
- All the FHE operations are publicly verifiable, allowing anyone to recompute the result and identify malicious FHE nodes. This is akin to optimistic rollup security, but for FHE computation. Furthermore, we don’t run a single FHE node, but rather have 3 operators run FHE nodes and sign their outputs, allowing to have both optimistic security and consensus on the result.

Archived copy · cryp2nova.com

- We use 13 nodes with a 2/3 majority rule for all our MPC protocols, while most other projects only use 3 to 5 nodes. Furthermore, our MPC protocol is robust, meaning it will give a correct output with up to 1/3 malicious nodes. As far as we know, this is the first implementation in production of a robust MPC protocol.
- Additionally, our MPC protocol runs inside AWS Nitro Enclaves, adding a layer of defense in depth and preventing access to the underlying share of the FHE private key from outside the protocol. The enclave also offers an attestation of the software version the MPC nodes are running, allowing the protocol to keep track of software updates. The combination of MPC and Nitro enclaves means recovering the shares and using them outside of the protocol would require AWS and multiple MPC nodes to collude.
- Genesis operators are highly reputable organizations with billions at stake through their non-Zama activities, whether as professional validators, infrastructure providers, businesses, or other. As they are all doxxed, anyone can see if they misbehaved. This brings economic security beyond their on-chain stake, as being caught misbehaving in the Zama Protocol will likely impact their other activities.
- Slashing is done via governance, allowing anyone to suggest a recourse if they identify bad behavior in an operator. This offers greater flexibility by allowing to capture edge cases and treat issues on a case per case basis.
- The Zama Protocol is being audited by Trail of Bits and Zenith. This is one of the largest audits of a crypto protocol, with over 34 audit-weeks spent on it so far.

CRYP2NOVA

Compliance

Building confidential applications often requires complying with local regulations. For example, financial institutions need to know who their customers are, verify that they are eligible to access specific financial instruments, that they are not blacklisted etc. Similarly, token issuers could give themselves the right to see the balances and transactions of their users, and comply offchain with existing AML / compliance tools used by traditional finance today.

Archived copy · cryp2nova.com

Contrary to many blockchain confidentiality protocols that puts the burden of compliance on the end-users, the Zama Protocol enables applications to define their own compliance rules directly within their smart contracts.

The ability to have “programmable compliance” is a key advantage offered by FHE, and means that the protocol itself has no say on who can access which encrypted value. Developers decide what is best for their applications, not the Zama Protocol.

Future Improvements

The Zama Protocol is the most advanced confidentiality protocol to date, and can already scale to address most blockchain use cases. Nonetheless, there are several areas of improvements we are working on to make it even more decentralized, secure and scalable. These typically rely on a combination of better hardware, better algorithms, and ZK-ifying everything:

Reaching 100k tps

- **New FHE techniques**: we are constantly inventing new FHE techniques that improve performance. We expect the base algorithms to improve by 10-20x over the coming years, similar to the performance gains ZK had in the past few years.
- **FHE ASICs**: we are working with several companies on accelerating FHE with dedicated hardware. The goal is to make FHE 100x-1000x faster using ASICs, in the same way Bitcoin mining or AI has been improved with dedicated hardware. We expect the first accelerators to be available in 2027-2028.
- **ZK-rollup Gateway**: the Gateway currently uses an optimistic rollup. Our goal is to move to a ZK rollup and improve the performance to support tens of thousands of transactions per second with a latency of less than 100ms.

Archived copy · cryp2nova.com

Making the KMS even more bullet-proof

- **ZK-MPC**: currently, all MPC protocols require a majority assumption on the nodes running the protocol. While this is fine in practice, in theory it enables MPC nodes to collude and provide an incorrect result. Our current design relies of AWS Nitro Enclaves to ensures the MPC nodes run the correct software, but this makes verifiability dependent on hardware security, which is suboptimal. To address this, we are working on adding ZK proofs to the MPC protocols, allowing anyone to verify that the individual contributions of MPC nodes are correct.
- **Large MPC committees**: MPC doesn't scale well: the more parties are involved, the slower it gets. As a result, most MPC protocols run with less than 10 nodes. While the Zama Protocol already uses more (13 nodes), it would be preferable to increase that number to a hundred, ensuring even more robustness and decentralization.

Enabling anyone to be an operator

- **Running MPC inside HSMs**: a major issue with MPC protocols is the need to trust the nodes with not leaking their share of the private key. This is typically done by using TEEs and a trusted committee of nodes. However this does not enable permissionless participation, as malicious attackers could try to break the TEE and access the secret in it. As an alternative, we are exploring how to run MPC inside HSMs such as those used by banks and other critical infrastructure.
- **ZK-FHE**: by proving the correctness of the FHE computation, it becomes possible to replace the Coprocessor consensus by a Proof-of-Work style protocol where anyone can compete to execute FHE operations, as long as they provide a proof that the result is correct. Right now, the overhead of ZK on top of the overhead of FHE makes this impractical, but our team is making good progress.

Making the protocol fully post-quantum

- **Post-quantum ZKPoK**: Zama's FHE and MPC technologies are already resistant to quantum computers. However, the ZKPoK is not (similar to most ZK-SNARKs). We are working on replacing it with a lattice-based ZK scheme that is post-quantum.

Archived copy · cryp2nova.com

- **Post-quantum signatures**: while we can make the Zama Protocol components post-quantum, signature schemes used by Host Chains are **not** currently post-quantum. We unfortunately do not have control over this, as it is up to the Ethereum, Solana and other L1/L2s communities to migrate to post-quantum signatures.

Operations and governance

The Zama Protocol uses Delegated Proof-of-Stake, with 18 operators running the protocol: initially 13 KMS nodes and 5 FHE Coprocessors (then more over time). They are chosen according to the following rules:

- genesis operators are selected based on reputation, DevOps experience and offchain value (equity, revenue, market cap, …). This enables bootstrapping security via reputation, as an operator with a large business value will likely lose customers if they get caught misbehaving in the Zama Protocol.
- we will progressively allow anyone to become a KMS or Coprocessor operator. To do so, they will first need to demonstrate they can reliably run a node in testnet, then stake at least 0.5% of the circulating $ZAMA supply. Every epoch (eg 3 months), the top KMS and Coprocessor operators by stake are selected to run the protocol for the next epoch.
- the active operators earn staking rewards in $ZAMA tokens, based on their role and stake.

CRYP2NOVA

The 18 genesis operators at the time of mainnet launch are:

- FHE nodes: Artifact, Blockscape, Luganodes, P2P and Zama
- KMS nodes: Conduit, DFNS, Etherscan, Figment, Fireblocks, InfStones, Layer Zero, Ledger, Omakase, Open Zeppelin, Stake Capital, Unit 410 and Zama

Archived copy · cryp2nova.com

Updates to the Zama Protocol have to be adopted by a majority of operators to be effective. This includes software updates, changes to the fees, adding support for a new Host Chain, etc. The only exception is pausing the protocol in case of an emergency and blacklisting spammers and sanctioned addresses, which any operator can do (however unpausing / de-blacklisting requires multiple coprocessors to be involved). In case of abuse, operators can get slashed. This ensures that the Zama Protocol has a swift mechanism to address critical issues, while incentivizing operators to behave honestly. Governance proposals can be found on Aragon↗.

The $ZAMA token

The $ZAMA token is the native token of the Zama Protocol. It is used for protocol fees and staking. It follows a burn and mint model, where 100% of the fees are burnt and tokens are minted to reward operators.

Fee model

Deploying a confidential app on a supported chain is free and permissionless. Furthermore, the Zama Protocol does not charge for the FHE computation, instead charging for:

- **Verifying ZKPoKs**. Each time a user includes encrypted inputs in a transaction, they need to pay a fee to the Zama Protocol to verify it.
- **Decrypting ciphertexts**. When a user wants to decrypt a ciphertext, they need to pay a fee to the Zama Protocol.
- **Bridging ciphertexts**. When a user wants to bridge an encrypted value from one chain to another, it needs to request it from the Zama Protocol and pay a fee.

The protocol fees can be paid by the end user, the frontend app or a relayer. As such, developers can create applications without their users ever needing to hold $ZAMA tokens directly.

Archived copy · cryp2nova.com

Protocol fees are paid with $ZAMA tokens, but are priced in USD. A price oracle regularly updates the $ZAMA/USD price on the Gateway, which updates the number of $ZAMA tokens paid for each protocol functionality. This has several advantages:

- it ensures protocol fees are proportional to usage and not dependent on speculation
- it creates predictability for users, developers and relayers, which can model their costs in USD rather than potentially volatile tokens.

Additionally, the Zama Protocol uses a cloud-like fee model: developers can subscribe to monthly plans to get reduced fees, up to 99% versus pay-as-you-go. A smart contract on the Gateway keeps track of subscriptions and debits developers accounts automatically.

The initial fee structure is as follows. It can be changed by operators as the protocol matures and usage patterns / infrastructure costs evolves:

- **ZKPoK verification**: from $0.5 to $0.005
- **Bridging**: from $1 to $0.01
- **Decryption**: from $0.1 to $0.001

Taking a confidential token transfer as an example:

- amounts and balances are encrypted
- there is typically 3 decryptions per transaction, one for each of the sender and receiver balances, and one for the final amount transferred, which will be set to 0 if the transfer failed.
- as such, the total cost would be, depending on the discount:
  - ZKPoK verification of encrypted amount: [$0.005 - $0.5]
  - Decryption of 2 balances + amount: 3 * [$0.001 - $0.1] = [$0.003 - $0.3]
  - Total cost: $0.008 to $0.8

Archived copy · cryp2nova.com

This model is designed to be affordable for large users and profitable for operators, regardless of market condition and price volatility. Eg a user interacting just once a month with confidential apps would pay less than 1$ / transaction, while a user interacting with a high-volume app such as a confidential stablecoin payment app or a wallet would pay less than 1 cent / transaction.

Staking rewards

Operators need to stake $ZAMA tokens to participate in running the protocol and receive the associated staking rewards. Tokens distributed as staking rewards are minted according to an inflation rate (5% initially), which can be changed via governance. The staking portal is at staking.zama.org↗

When rewards are distributed, they are first split by role (coprocessors, KMS nodes), then distributed pro-rata of the square root of the stake of each operator within that group. Rewards can be claimed any time, but the stake itself has an unbonding period of 7 days. Distributing rewards this way ensures that each operator gets rewarded according to the job they did, while avoiding concentration of rewards into a few operators only.

Token holders can help secure the network by delegating their tokens to the operators of their choice and share their rewards (minus the operator's commission). When staking on an operator, delegators receive a liquid token representing their share of the staking rewards of that operator.

The table below summarizes the percentage of rewards going to each group, and the expected operator infra costs:

Role	% of rewards / operator	Number of operators	Monthly infra cost / operator
Coprocessors	8%	5	$15,000 / 10 tps on host chains
KMS	4.6%	13	$5,000 / 50 tps decryptions

Archived copy · cryp2nova.com

Distribution

20% of the supply will be circulating at TGE, with the rest subject to lockups as follows:

Search

Group	Allocation	Lockup	Cliff	Unlock at Cliff
Public Sale	12%	No lockup	-	-
TGE Campaigns	6%	No lockup	-	-
Liquidity	2%	No Lockup	-	-
Treasury	20%	2 years	None	-
Growth	10%	4 years	None	-
Team	20%	4 years	1 year	1/10th
VCs	20%	2 years	1 year	1/10th
Angels	10%	2 years	1 year	1/10th

CRYP2NOVA

Archived copy · cryp2nova.com

About Zama, the company

The Zama Protocol is a spinout from Zama, an open source cryptography company building state-of-the-art Fully Homomorphic Encryption (FHE) solutions for blockchain and AI.

Zama has raised over $150m at a $1.2b valuation from some of the most successful blockchain investors, including Multicoin, Pantera, Blockchange and Protocol Labs, as well as founders of major protocols such as Juan Benet (IPFS/Filecoin), Gavin Wood (Ethereum/Polkadot), Anatoly Yakovenko (Solana), Sandeep Nailwal (Polygon), and others.

CRYP2NOVA

Team

Zama is a cryptography company operating across the globe. It was founded in 2020 by Dr Rand Hindi (CEO) and Dr Pascal Paillier (CTO), with other prominent researchers leading the company, such as Prof Nigel Smart (Chief Academic Officer) and Dr Marc Joye (Chief Scientist). There are more than 90 people working at Zama, of which nearly half hold PhDs, making Zama the largest research team in FHE.

About the founders:

Archived copy · cryp2nova.com

- **Rand** is an entrepreneur and deeptech investor. He is the CEO at Zama and a partner at Unit.vc ↗, where he invested in over 100+ companies across cryptography, AI and biotech. Rand is also a competitive biohacker, and currently ranks in the top 5% of the Rejuvenation Olympics with an aging rate of 0.68. Rand started coding at the age of 10, founded a Social Network at 14 and started his PhD when he was 21. He then created Snips, a confidential AI startup that was acquired by Sonos. He was previously a member of the French Digital Council, advising the government on AI and Privacy issues, a lecturer at Science Po University in Paris, and an advisor to several biotech, AI and defense companies. He holds a BSc in Computer Science and a PhD in Bioinformatics from University College London (UCL).
- **Pascal** is a pioneer in FHE and cryptography, and the CTO at Zama. He invented one of the first additive homomorphic scheme (the Paillier encryption scheme ↗), which is still widely used today. Pascal has published dozens of papers, with major contributions across various cryptography domains, including FHE, smart cards, and more. Prior to Zama, he led the cryptography innovation team at Gemalto, and founded CryptoExperts a leading cryptography consulting firm. Pascal is a 2025 IACR fellow, received several awards for his research, and led multiple ISO standards for cryptography. He holds a PhD in cryptography from Telecom Paris.

Products & Services

Everything we do is open source under a dual licensing model. It is free for non-commercial use, prototyping, research and personal projects, but commercial use requires either obtaining an enterprise license or building on top of a protocol that already has one.

> ⓘ Developers building on the Zama Protocol don't need an extra license. However, forking, copying or using Zama's technology outside of the Zama Protocol does require a license.

We offer several products and services:

- **FHE libraries** for AI and blockchain. This includes TFHE-rs ↗, FHEVM ↗, Concrete ML ↗, and TKMS ↗. They are free for non-commercial use, but require an enterprise license for commercial use.

Archived copy · cryp2nova.com

- **Hosted services**, such as an encryption/decryption relayer and a decryption oracle, that make it easy for app developers to use the Zama Confidential Blockchain Protocol and other protocols based on our FHEVM technology.
- **Premium support** for companies and developers who need help building and managing their FHE applications.

There are over 5,000+ developers using our libraries, representing a 70% market share. Our technology has furthermore been licensed to dozens of companies, including L1s, L2s, finance and AI. Nearly all decentralized protocols using FHE are using Zama's technology behind the scene.

> Note that the Zama Protocol is operated as an independent, decentralized protocol. The services we offer on the company side are independent of the protocol itself, and are meant to serve enterprises and developers who want to build confidential applications, regardless of whether they are deployed on the Zama Protocol or not.

Additional links

- Zama Protocol docs ↗
- FHEVM whitepaper ↗
- TFHE-rs handbook ↗
- Zama GitHub ↗
- Discord ↗
- X ↗
- Zama blog ↗

CRYP2NOVA

Archived copy · cryp2nova.com

Disclaimer

The present light paper and/or any other accompanying documentation ("**Document**") only provide educational material about the Zama Protocol and the $ZAMA token. Please note that the Zama Protocol and the $ZAMA token are under active development and are subject to change. Zama may change this Document at any time at its sole discretion without notice.

Any documentation is provided for informational purposes only and does not constitute some kind of prospectus, key information document, or similar document. No prospectus, key information document, or similar document will be provided at any time. There is no guarantee for the completeness of the documentation provided. All numbers and forward-looking statements mentioned within the present document as well as any accompanying documentation reflect mere estimations/indications. They are not guaranteed and may change substantially.

Any and all liability of ZAMA Switzerland AG and/or any affiliated legal entity or private individual for the completeness and accuracy of the documentation provided and any damages arising from reliance on such documentation is limited to the fullest extent permitted by any applicable law.

Any dispute related to or arising out of the information provided within the present Document as well as any accompanying documentation shall be submitted to the exclusive jurisdiction of the competent courts of Zug, Switzerland, with the exclusion of any other jurisdiction or arbitration.

This disclaimer, the Document, as well as any accompanying documentation shall be governed by and construed and interpreted in accordance with the substantive laws of Switzerland, excluding the Swiss conflict of law rules.

Last updated 6 months ago

Archived copy · cryp2nova.com

Archived copy · cryp2nova.com