PGNChain.com

Private on-chain wallet to wallet messaging and hidden payments protocol Near Zero Cost

By Lizeth Sullivan & Lilliam Carr
PGNChain Foundation
13 January 2025

Abstract

PGNChain is a decentralized, privacy-focused protocol that enables encrypted wallet-to-wallet messaging and claim-based fund transfers on-chain. Unlike existing communication tools that operate off-chain or expose sensitive data in plaintext, PGNChain leverages asymmetric encryption and decentralized storage to provide secure, verifiable communication without compromising user privacy.

Messages are encrypted off-chain using recipients' wallet-based public keys, stored on distributed file networks like IPFS or Arweave, and referenced optionally through minimal on-chain logs. In parallel, PGNChain supports value transfer via vault-based deposits, where funds are claimed using cryptographic secrets shared through encrypted messages — eliminating public sender–recipient links.

The protocol is chain-agnostic, lightweight, and modular — deployable on any EVM-compatible blockchain, and designed to function under $0.01 per interaction. PGNChain does not introduce a new blockchain, but instead acts as a messaging-and-settlement layer over existing infrastructure. This makes it ideal for DAOs, DeFi protocols, NFT projects, and privacy-conscious users seeking a trustless and verifiable way to communicate and exchange value.

Content

Introduction

While decentralized finance, NFTs, and DAOs have advanced the economic and social layers of Web3, on-chain communication remains underdeveloped, fragmented, and largely dependent on centralized services. Wallets today cannot directly message one another. Conversations surrounding governance, coordination, and contract negotiation occur off-chain, often on platforms like Discord or Telegram—tools that are fundamentally ill-suited for secure, trustless, and verifiable interaction. These environments break the principle of decentralization and expose users to privacy risks, impersonation, surveillance, and censorship.

Blockchain transactions are public, permanent, and pseudonymous, yet there is no reliable way to transmit an encrypted message alongside value, let alone establish a reply channel, verification mechanism, or time-sensitive exchange. The lack of native messaging infrastructure has forced Web3 coordination into off-chain silos, which disconnects identities from assets and actions. Existing solutions either attempt messaging through inefficient on-chain storage, rely on Web2 services, or omit privacy and encryption entirely.

PGNChain introduces a fundamental communication primitive to blockchain: encrypted wallet-to-wallet messaging combined with optional value delivery. It allows users to craft messages that are encrypted off-chain using the recipient's public key, stored on decentralized file systems like IPFS or Arweave, and optionally anchored to the chain via minimal data references. Alongside the messaging layer, PGNChain also provides claim-based fund transfers that eliminate the need for public sender-to-recipient linkage. This is achieved through vault-based deposits secured by cryptographic secrets, which can only be revealed and claimed by the intended recipient.

The protocol does not propose a new blockchain or consensus system. Instead, it is designed as a lightweight, chain-agnostic layer that integrates seamlessly with existing EVM-compatible ecosystems such as Ethereum, Base, Polygon, and BNB Chain. Future implementations may include Solana and Cosmos, as well as OP_RETURN-style payload encoding for limited-use Bitcoin interoperability.

PGNChain offers composability without friction. It is designed to be modular, interoperable, and developer-friendly—enabling any application to embed trustless communication, encrypted signaling, and micro-settlement capabilities. Whether enabling private DAO voting, airdrop access control, dispute resolution, or stealth gifting, PGNChain serves as the secure communication backbone that Web3 has been missing.

Design Principles

PGNChain is built on the foundational belief that private communication and verifiable coordination should be native capabilities of decentralized systems—not external add-ons. The protocol is architected around a set of principles that prioritize privacy, minimalism, cost-efficiency, and universal composability. At its core, PGNChain is an off-chain encrypted messaging and on-chain delivery framework that seeks to fill the gap left by traditional blockchains, which were never designed for direct, encrypted human interaction.

The protocol treats encryption as a default, not an option. Every message sent via PGNChain is encrypted using asymmetric cryptography, relying on the recipient's public wallet key to ensure that only the intended party can decrypt it. This encryption happens entirely off-chain, preserving blockchain space and reducing costs, while ensuring that even metadata like sender identity and message intent remain hidden unless explicitly disclosed by the recipient. Decryption takes place locally, using the user's wallet-integrated private key, which means that no off-chain server, relay, or third party can intercept or read the message content.

Cost minimization is a key architectural goal. PGNChain avoids bloating the blockchain with full message payloads. Instead, it stores only compact message references, such as hashed pointers or CIDs from IPFS or Arweave, within lightweight smart contracts. By removing the need for heavy state changes or full message replication, the protocol achieves a cost profile that remains under a cent per interaction, even on moderately priced Layer 2s.

Chain-agnostic deployment is another guiding principle. PGNChain is not tied to a single ecosystem. It is designed to operate across any EVM-compatible chain, with future support planned for non-EVM environments. This ensures maximum reach and interoperability while giving developers and users the flexibility to interact within their preferred blockchain environments.

Finally, PGNChain is built as a protocol, not a platform. It does not impose UI or centralized coordination but instead provides a minimal set of primitives—encryption libraries, message registries, claimable vaults, and optional relayer contracts—that can be composed, integrated, or extended by any developer. The goal is to allow applications, wallets, DAOs, and protocols to embed encrypted communication and conditional fund transfer functionality as easily as embedding a transaction button or signing a message.

By adhering to these design principles—default encryption, off-chain storage, minimal on-chain footprint, cross-chain compatibility, and modular integration—PGNChain delivers a trustless communication layer that behaves like a true protocol: invisible when needed, powerful when used, and extensible without compromise.

Architecture Overview

The architecture of PGNChain is deliberately minimalist yet deeply modular. It is designed to enable encrypted communication and value settlement without introducing unnecessary layers, consensus mechanisms, or infrastructure burdens. The protocol leverages off-chain cryptographic operations, decentralized storage networks, and lightweight on-chain smart contracts to ensure both scalability and security while minimizing cost. Every component is designed to operate independently or in concert, making the system highly composable and adaptable to a wide variety of blockchain environments and applications.

At the heart of the system lies the encryption engine, which uses elliptic curve cryptography to encrypt messages before they are ever touched by the chain. This engine is implemented client-side, typically within a browser or wallet extension, and uses the recipient's public key to generate a secure, asymmetric payload. Once encrypted, the message is uploaded to a decentralized file storage network, such as IPFS or Arweave, where it becomes publicly retrievable but cryptographically unreadable by anyone other than the intended recipient. The resulting content identifier (CID) or hash acts as a pointer to the message, and it may be optionally anchored to the blockchain using a smart contract that serves as a minimal message log. This anchoring is not required for every message but provides a trustless, time-stamped record when verifiability is necessary.

For value transfer, PGNChain introduces a claim-based fund delivery mechanism called the PigeonVault. In contrast to traditional transfers, where funds are sent directly from one address to another, the PigeonVault allows users to deposit assets into a shared on-chain pool without specifying a recipient. Instead, access to the funds is controlled by a secret—typically a hash commitment—that is shared privately through an encrypted message. The recipient can later use this secret to claim the funds from the vault, allowing for private, verifiable transfers that do not publicly link sender and recipient on-chain.

To support messaging and claims without requiring users to pay gas fees in every scenario, PGNChain also supports optional relayer contracts. These relayers can be operated by third parties or protocol validators who are compensated in PGN tokens for submitting meta-transactions or performing off-chain indexing and inbox aggregation. This model ensures accessibility for low-balance users while preserving decentralization.

PGNChain does not depend on any specific blockchain and is fully chain-agnostic. The message log and vault contracts can be deployed independently on any EVM-compatible network, and the protocol can operate simultaneously across multiple chains. Because message encryption and decryption are handled entirely off-chain and storage is decentralized, the protocol's logic and data are not bound to a single execution environment. This cross-chain capability makes PGNChain ideal for wallets, DAOs, and protocols that want to offer private communication or stealth transfers without changing their underlying blockchain stack.

Together, the encryption engine, decentralized storage, lightweight on-chain logs, claimable vault system, and optional relayer infrastructure form a robust but low-overhead protocol. This architecture achieves the goal of enabling encrypted on-chain messaging and value delivery in a way that is trustless, scalable, and composable from day one.

Encryption Model

The PGNChain encryption model is rooted in asymmetric cryptography and designed to provide strong confidentiality guarantees without requiring any interaction between the sender and recipient beyond a public wallet address. Unlike traditional messaging platforms that rely on shared passwords or centralized key management, PGNChain allows anyone to encrypt a message for any wallet address by using the recipient’s existing public key derived from their on-chain identity. This ensures that all encrypted payloads are inherently compatible with wallet-based systems and do not require users to manage additional key pairs or credentials.

When a user wishes to send a message, the PGNChain client generates an ephemeral session key and encrypts the message content using a combination of elliptic curve Diffie–Hellman (ECDH) and symmetric encryption such as AES. The public key of the recipient, which can be derived from their on-chain activity or via EIP-1024 mechanisms, is used to compute a shared secret. This secret serves as the basis for encrypting the message, ensuring that only the corresponding private key—controlled by the recipient's wallet—can decrypt it. The encryption is performed locally, entirely off-chain, and without any server-based intermediaries, making the process highly secure and resistant to interception.

Once encrypted, the message is uploaded to a decentralized storage layer, typically IPFS or Arweave, where it is assigned a unique content identifier. This identifier may be transmitted to the recipient through an on-chain reference, such as an event emitted by a PGNChain smart contract, or distributed entirely off-chain through an alternative channel. The CID itself carries no semantic information and is unintelligible without the decryption key, providing strong metadata privacy even when the message is publicly accessible.

On the recipient side, decryption is performed by the PGNChain-enabled wallet or DApp interface. When a user loads the encrypted message, the client prompts their wallet to compute the shared secret using their local private key. This decryption process happens entirely within the user's browser or mobile environment, and no unencrypted data is ever transmitted externally. The use of wallet-native keys ensures that decryption does not require any manual key entry, credential exchange, or password setup, offering both high usability and robust security.

The protocol also supports optional extensions to the encryption model. These include password-based encryption for shared secrets, NFT-gated access to messages, and time-locked decryption schemes where messages can only be accessed after a certain block height or timestamp. Such features add flexibility for applications that require more complex access control, such as DAOs managing governance communication or teams distributing confidential airdrop links.

Although the default implementation relies on ECC-based encryption, PGNChain is structured to accommodate future enhancements such as post-quantum encryption or zero-knowledge proof-based message authentication. These upgrades can be integrated without breaking backward compatibility, thanks to the off-chain nature of the encryption layer and the modularity of the message format.

In summary, PGNChain's encryption model delivers secure, wallet-native, asymmetric message confidentiality that is easy to integrate, resistant to surveillance, and adaptable to evolving privacy needs. It forms the cryptographic foundation for all messaging and value transfer interactions in the protocol.

Message Lifecycle

The lifecycle of a message within PGNChain follows a carefully structured process that begins with the sender's intention to communicate and culminates in a secure, trustless interaction that optionally involves a value transfer. This lifecycle is designed to ensure that every message is encrypted, verifiable, and private from end to end, while offering flexibility for developers and users to decide how and where message metadata is anchored on-chain.

The process starts with the sender preparing a message intended for a specific recipient's wallet address. Using the PGNChain client or wallet interface, the sender composes the message and initiates the encryption process. The recipient's public key is used to derive a shared encryption key via elliptic curve Diffie–Hellman key exchange, and the message is then encrypted using symmetric encryption such as AES. This encryption occurs entirely off-chain within the sender's client environment, ensuring that the plaintext message is never exposed to the network, intermediaries, or storage layers.

Once encrypted, the message is uploaded to a decentralized file storage network such as IPFS or Arweave. These networks provide a content-addressable identifier, commonly known as a CID, which uniquely references the encrypted payload. The CID is public, but because the message is encrypted, the contents remain confidential. At this stage, the sender may choose to transmit the CID to the recipient in several ways. If on-chain verifiability is desired, the sender can call a PGNChain smart contract to store the CID hash in an event log or minimal message registry. If privacy is paramount and verifiability is not needed, the sender can distribute the CID through any off-chain communication channel, such as email, social messaging, or QR code, or embed it within another smart contract interaction.

The recipient, upon becoming aware of the message, either by observing the on-chain message registry or through off-chain delivery, retrieves the encrypted content using the CID. Using their wallet-integrated PGNChain client, the recipient decrypts the message locally. The client uses the recipient's private key to reconstruct the shared encryption key and decrypt the payload, thereby revealing the message. This process ensures that only the intended recipient, and no external observer or indexer, can read the contents of the message.

In cases where the message includes a reference to claimable funds, such as a hash preimage or secret key for PigeonVault, the recipient can use this information to trigger a value claim within a PGNChain smart contract. This enables seamless transition from communication to transaction without exposing the linkage between sender and recipient on-chain. Moreover, the recipient can craft a response to the original sender using the same mechanism in reverse—encrypting a new message using the sender's public key, storing it via the same off-chain infrastructure, and optionally anchoring the reference through the protocol's messaging contract.

The entire lifecycle—from encryption to delivery, decryption, and optional reply—operates without requiring any centralized server, custody of user keys, or assumption of trust between participants. Every step is independently verifiable, and each message can exist as either a private exchange or a public proof of interaction, depending on how the sender chooses to handle the storage reference. In this way, PGNChain provides a flexible, trust-minimized communication lifecycle that mirrors the simplicity of messaging while embracing the integrity and resilience of decentralized infrastructure.

Fund Delivery via PigeonVault

In addition to encrypted messaging, PGNChain enables private and verifiable value transfer through a system called PigeonVault. Unlike traditional token transfers, which directly link sender and recipient addresses in a transparent transaction, the PigeonVault introduces a claim-based mechanism that obscures the recipient's identity while still ensuring trustless, cryptographically enforced settlement. This approach eliminates the need for direct address-to-address transfers and creates a secure and private method for distributing funds on-chain.

The fund delivery process begins with the sender depositing tokens into the PigeonVault smart contract. During this deposit, the sender specifies a cryptographic commitment in the form of a hashed secret. This secret acts as a claim key and is shared with the intended recipient through an encrypted message sent via PGNChain's messaging infrastructure. The contract does not require or even record the recipient's wallet address, ensuring that no publicly viewable association is created between the sender and the eventual claimant. The only on-chain information stored is the hash of the secret, the amount of tokens locked, and an optional expiration time or burn condition.

Once the recipient receives the encrypted message containing the secret or claim code, they can use this information to interact with the PigeonVault and prove their entitlement to the funds. The vault contract verifies the submitted secret by hashing it and checking for a match with the stored commitment. If the hash matches, the contract releases the locked tokens to the claimant's address. Because the vault does not know or enforce a predefined recipient, anyone with the correct secret may claim the funds, but only the intended recipient would have received the claim key, making the system both permissionless and privacy-preserving.

This model allows for highly flexible value delivery use cases. A sender can create pre-funded messages that act like claimable coupons, gift tokens to an unknown recipient in advance, or create time-locked incentives that can only be claimed after a certain period. Additionally, the claim mechanism can be enhanced with optional features such as stealth withdrawals, where the funds are routed to newly generated addresses for maximum unlinkability, or post-claim token burns that erase all metadata associated with the transaction, ensuring maximum transactional anonymity.

In terms of cost efficiency, the PigeonVault is optimized for low gas consumption. The only on-chain writes occur during the initial deposit and the final claim, both of which involve simple data storage and verification steps. The logic is deterministic and does not rely on third-party oracles or off-chain validators, which maintains simplicity and verifiability while avoiding additional infrastructure complexity.

PigeonVault represents an evolution in how value is moved across wallets. It shifts the paradigm from push-based transfers to pull-based, claim-driven delivery. Combined with encrypted messaging, it allows users to send both information and value privately, securely, and with optional traceability. This mechanism makes PGNChain uniquely capable of handling scenarios like stealth airdrops, private compensation, incentive disbursement, and on-chain inheritance — all without exposing the financial trail or relying on centralized coordination.

Smart Contract Modules

PGNChain operates through a set of lightweight, purpose-specific smart contracts that together enable encrypted communication, on-chain message referencing, and claim-based fund transfers. These modules are intentionally minimal in design to reduce gas consumption, simplify auditing, and allow for modular deployment across multiple blockchains. Each contract is self-contained, yet interoperable, allowing developers to integrate only the components they need while maintaining the protocol's core principles of privacy, verifiability, and decentralization.

The first core contract is the MessageLog. This contract acts as a public ledger of encrypted message references, such as IPFS content identifiers or hashed payload pointers. When a user sends a message, they may choose to record a reference to the encrypted blob by emitting an event or writing to a minimal registry within the MessageLog. The contract does not store the message content itself, nor does it reveal sender or recipient identities unless the sender chooses to log a public identifier. This allows messages to be timestamped and verifiable on-chain without revealing any private data. MessageLog can be used by DAOs, dispute systems, or any application that requires cryptographic proof of a message having been sent at a specific time.

The second contract, and perhaps the most critical for value transfer, is the PigeonVault. This vault allows users to deposit tokens or other digital assets with a secret hash commitment. The contract records the committed hash and holds the funds until someone submits a valid pre-image of the hash — in other words, the secret required to claim the deposit. Importantly, PigeonVault does not bind funds to any specific recipient address, which removes any on-chain evidence linking sender and receiver. This is especially useful in contexts that require privacy, such as anonymous tipping, stealth funding, or encrypted airdrops. The contract includes optional expiration parameters, allowing funds to be refunded to the sender or burned entirely if unclaimed within a predefined timeframe.

To further enhance accessibility and user adoption, PGNChain supports the use of relayer contracts as a third optional module. These relayers enable users to interact with PGNChain without paying gas directly, through the use of meta-transactions signed off-chain. The relayer receives a signed payload from the user, broadcasts the transaction on their behalf, and optionally receives compensation in PGN tokens. This design allows the protocol to be used even by wallets that hold no native chain tokens, dramatically lowering the barrier to entry and enabling seamless interaction for low-balance users or first-time participants.

Each of these smart contracts is optimized for modular deployment. Developers can deploy the MessageLog and PigeonVault independently or together, depending on the desired functionality. This ensures PGNChain can scale horizontally across EVM-compatible chains without requiring a centralized bridge or coordination layer. Contracts are written to be fully upgradeable via standard proxy patterns if desired, or immutable when fixed security guarantees are required.

In its smart contract architecture, PGNChain balances minimalism and power. The contracts are small enough to be deployed inexpensively on most Layer 2 networks, yet expressive enough to support a wide range of use cases. By abstracting core primitives — message logging, encrypted referencing, secret-based claiming, and gasless access — PGNChain establishes itself as a protocol-level messaging and settlement layer that can be embedded, extended, and trusted across the decentralized ecosystem.

Cross-Chain Support

PGNChain is designed from the ground up to be chain-agnostic, with native support for deployment across multiple blockchain networks without dependence on a single Layer 1 or Layer 2. Its architecture separates the critical components of encryption, storage, and verification in such a way that they can operate independently of any specific chain's consensus or execution model. This makes PGNChain not only scalable and interoperable, but also adaptable to the shifting dynamics of the multi-chain Web3 ecosystem.

The protocol's encryption and storage processes are inherently off-chain, which means the most sensitive operations — such as generating shared secrets and encrypting message content — are decoupled from the chain itself. These encrypted messages are stored on decentralized networks such as IPFS or Arweave, where they can be retrieved by any wallet or application, regardless of the user's blockchain environment. The message payloads remain accessible and verifiable without requiring cross-chain bridges or token transfers. Only the content-addressable identifiers, or CIDs, may be optionally anchored to a given chain's messaging registry to provide timestamped proof of communication.

Smart contracts that support PGNChain's operations — including the MessageLog and PigeonVault — can be deployed on any EVM-compatible network. This includes major chains like Ethereum, Polygon, BNB Chain, Arbitrum, Optimism, and Base. Each deployment operates independently, meaning users can send messages and fund claims entirely within the context of their chosen chain. However, because the encryption and content storage are off-chain, it is also possible for a message composed on one chain to be decrypted and interacted with by a recipient on another chain, provided that the reference to the content is shared and accessible. This enables a form of lightweight cross-chain communication without requiring the complexity or security risks of traditional bridging.

For more advanced scenarios, PGNChain can integrate with existing interoperability protocols such as LayerZero, Axelar, or IBC-compatible relayers to facilitate automatic propagation of message references or claim proofs across chains. These systems can be used to synchronize claim rights or to broadcast message pointers to multiple chains simultaneously, allowing developers to build universal inboxes, interoperable DAO messaging layers, or cross-chain reward distribution mechanisms that remain verifiable regardless of execution environment.

Because PGNChain separates the communication logic from the execution environment, the same protocol can also be extended to non-EVM chains with relative ease. In the future, Solana-based implementations may rely on message logs stored in account data, while Cosmos zones can use IBC-enabled modules to handle vault commitments and claim verification. Even Bitcoin can be included in the ecosystem through OP_RETURN transactions that publish encrypted references or claim secrets, enabling a form of encrypted, verifiable signaling between Bitcoin wallets and other smart contract platforms.

Ultimately, PGNChain's cross-chain support is not based on complex infrastructure or token bridging but on simplicity, modularity, and a strong separation between encryption, storage, and execution. This enables true composability and interoperability, allowing encrypted messages and value flows to move freely across the blockchain universe while maintaining privacy, integrity, and user control.

Token Model: $PGN

At the heart of PGNChain's economic architecture lies the $PGN token, a native utility and governance asset designed to power protocol-level interactions while incentivizing usage, decentralization, and long-term sustainability. The $PGN token is not simply a means of payment; it is an access key, a coordination layer, and a privacy enabler that aligns the incentives of users, developers, relayers, and ecosystem participants within a trustless messaging and settlement framework.

The core utility of $PGN is rooted in message anchoring and fund claim verification. While encrypted messages can be transmitted entirely off-chain, many applications require on-chain verifiability. To store message references in the PGNChain MessageLog or to commit claimable deposits within PigeonVault, users may pay a small fee denominated in $PGN. These fees are deliberately kept low to preserve the near-zero cost nature of the protocol but serve an important role in preventing spam, ensuring fair usage, and sustaining the network. In environments where native gas tokens are unavailable or impractical, $PGN can be used as a universal fee token, abstracting away the need to hold multiple assets across different chains.

Beyond its role in message and vault interactions, $PGN is also central to the privacy-preserving features of the protocol. Users who wish to obscure their on-chain message trails can elect to burn $PGN when submitting references, removing sender metadata and increasing the unlinkability of interactions. This optional burn function operates as a privacy enhancement mechanism, enabling stealth messaging or anonymous value transfers in a provable, irreversible manner.

Staking is another integral function of the token. $PGN holders may stake tokens to operate or delegate to relayers — nodes or smart contracts that submit signed meta-transactions on behalf of gasless users. These relayers earn $PGN in exchange for their services, creating a circular economy that encourages infrastructure growth and user onboarding. As relayers become more distributed, the network gains resilience and censorship resistance, aligning with PGNChain's broader decentralization goals.

In the future, $PGN may also serve as a governance token, allowing token holders to vote on protocol upgrades, parameter changes, privacy policies, or incentive structures. Governance can be implemented using token-weighted voting or more advanced quadratic or reputation-based systems, depending on community evolution and adoption scale. The introduction of decentralized governance ensures that the protocol remains adaptable to new privacy standards, cryptographic techniques, and evolving use cases without relying on a centralized authority.

From a tokenomics perspective, $PGN is designed with a fixed total supply, ensuring long-term scarcity and value alignment. A portion of the initial supply is allocated to ecosystem development, protocol grants, staking rewards, and community airdrops to bootstrap early adoption. Contributor and team allocations are subject to vesting schedules to ensure alignment with the protocol's growth and utility milestones. No uncontrolled emissions or inflation mechanisms are planned, as the protocol is designed to be self-sustaining through lightweight fee capture and value-added services.

In summary, $PGN is more than just a token. It is the connective tissue that enables communication, coordination, and confidentiality across the PGNChain ecosystem. Whether used to log a message, verify a claim, obfuscate a sender, power a relayer, or participate in protocol governance, $PGN provides the economic backbone of a messaging and micro-settlement layer purpose-built for Web3.

Use Cases

PGNChain introduces a foundational capability to the blockchain ecosystem: encrypted, verifiable, and composable communication and fund transfer. Its architecture supports a wide range of practical applications across DeFi, DAOs, NFT ecosystems, and privacy-driven user experiences. These use cases emerge organically from the combination of off-chain encryption, decentralized storage, and on-chain claimable value, creating opportunities for trustless coordination that have previously required centralized infrastructure or invasive visibility.

One of the most immediate applications of PGNChain is in the realm of decentralized autonomous organizations. DAOs today rely on external platforms to manage internal discussions, proposal negotiations, and coordination between stakeholders. By integrating PGNChain, DAOs can enable encrypted wallet-to-wallet messaging between members, allowing sensitive conversations, confidential proposals, or multisig coordination to occur directly within the crypto-native environment. With optional message logging, the DAO can decide whether to preserve auditable records or maintain full communication privacy, enabling nuanced governance structures that adapt to the nature of the decision being made.

PGNChain also unlocks a new dimension for airdrops and community rewards. Traditional airdrops are noisy, often public, and offer no privacy to recipients. By leveraging encrypted messages paired with PigeonVault claims, projects can create targeted, stealth airdrops where only the intended wallet holder receives the claim key. This model also allows conditional or interactive airdrops, where the recipient must respond to a message, complete a verification step, or engage with the community before accessing the reward. These mechanics increase participation quality while preserving privacy and reducing Sybil exploitation.

Another compelling use case is crypto gifting and private payments. PGNChain allows users to send tokens to a recipient without revealing either party's identity on-chain. The sender encrypts a message with a vault claim secret and transmits it directly to the recipient. The recipient decrypts the message and claims the funds. There is no on-chain link between the addresses, making it ideal for one-time payments, anonymous donations, or gifting digital assets to users who do not yet hold any tokens. This model also enables crypto-native gift cards or redemption codes that can be transmitted securely and claimed trustlessly.

In the realm of NFT ecosystems, PGNChain enables private messaging between collectors, creators, and marketplaces. Creators can send encrypted updates, unlockable content, or personalized messages to verified NFT holders without exposing the conversation publicly. Collectors can respond securely, propose trades, or coordinate in-wallet collaborations. These interactions can be linked to NFT ownership at the time of message delivery, ensuring that only legitimate holders can decrypt or act on the message content, while providing a direct and private communication line between wallets.

The protocol also offers meaningful utility in dispute resolution, arbitration, and legal coordination. Smart contract-based systems often lack nuanced communication layers, forcing participants to exit the protocol to negotiate terms or resolve disputes. With PGNChain, arbitrators or participants can exchange encrypted messages that are timestamped, verifiable, and optionally auditable without exposing the conversation to the entire network. This supports more mature decentralized systems, where economic logic must be paired with trusted communication channels.

Finally, PGNChain opens the door to decentralized inheritance and time-based value release. A user can create a message encrypted with a vault claim code and store the message publicly while keeping the claim secret time-locked or hidden. When a specified condition is met — such as a timestamp, a smart contract event, or a delegated action — the secret can be revealed, enabling a future claimant to retrieve the value. This creates a mechanism for on-chain wills, delayed payouts, or confidential unlocks that do not rely on any custodial party or external legal system.

In all of these use cases, PGNChain acts not just as a messaging tool but as an embedded coordination layer. It enables programmable communication that is encrypted, verifiable, and seamlessly linked to value transfer — expanding what is possible in decentralized systems by bringing human interaction into the trustless design space.

Security & Privacy Considerations

PGNChain is fundamentally designed around the principles of privacy, non-interactivity, and cryptographic verifiability. Each element of the protocol has been engineered to preserve confidentiality, minimize data exposure, and provide strong guarantees against unauthorized access or metadata leakage. At the same time, it adheres to the transparency expectations of blockchain systems, allowing users and developers to verify interactions when needed, while preserving the optionality of stealth and unlinkability.

The protocol's security model begins with the encryption process itself. Messages are encrypted off-chain using asymmetric cryptography based on elliptic curve standards, specifically secp256k1, which aligns with the cryptographic primitives used by most blockchain wallets. This ensures compatibility while maintaining high levels of security against brute force decryption or interception. Because the encryption is performed locally within the sender's client, there is no exposure of message content to any server, relay, or middleware. Decryption similarly occurs on the recipient's device, using their wallet's private key, and does not require the key to be extracted or exposed. This end-to-end encryption model provides robust confidentiality even in adversarial network environments.

The storage layer, typically IPFS or Arweave, plays a crucial role in preserving data availability while avoiding centralization risks. Although these networks are publicly readable, the encrypted nature of the message payloads ensures that no information is leaked through content access. To prevent timing correlation or scraping attacks, messages can be optionally delayed, batched, or sent through proxy services or relayers that obfuscate upload origins. Because the protocol does not rely on any centralized storage or index, users maintain full control over their content's lifecycle, and adversaries are unable to infer relationships from access patterns alone.

One of the most important privacy guarantees in PGNChain comes from the claim-based fund delivery mechanism. Unlike traditional transfers, where the sender and recipient addresses are permanently linked on-chain, the PigeonVault allows funds to be deposited anonymously with only a hashed secret recorded. When a recipient claims the funds, the smart contract verifies the secret without ever needing to know the intended recipient's address. This architecture eliminates one of the most common sources of metadata leakage in crypto: the direct association between payment sender and receiver. For further privacy, the recipient may choose to claim the funds using a freshly generated stealth address, ensuring no backtracking is possible even with advanced chain analysis tools.

PGNChain also supports optional burn mechanics and metadata minimization techniques. Users who wish to maximize privacy may choose to burn PGN tokens when submitting message references, removing sender fingerprints and reducing the chain footprint of their interaction. Contracts are designed to emit minimal logs and avoid address mappings unless explicitly required by the application logic. Where message anchoring is used, it can be stripped of sender metadata or substituted with zero-knowledge proofs that verify intent or timing without revealing the underlying identities or content.

To guard against denial-of-service attacks or spam flooding, PGNChain includes lightweight rate-limiting mechanisms and fee structures. Posting to the MessageLog or depositing to the vault requires nominal fees, which deter abuse without compromising accessibility. Additionally, protocol relayers who submit messages on behalf of users via meta-transactions are subject to signature verification and nonce protection to prevent replay attacks or signature reuse.

Future iterations of the protocol may incorporate zero-knowledge proof systems to further enhance anonymity and allow for private claims, message attestation, and sender verification without disclosing any sensitive information. These systems can be integrated modularly, allowing applications to choose the appropriate balance between privacy and performance. Post-quantum cryptographic support is also a long-term consideration, ensuring that the protocol remains secure against evolving threats while maintaining compatibility with existing wallet infrastructure.

In sum, PGNChain is built to deliver privacy by design, without sacrificing the trustlessness and auditability that define blockchain systems. By combining end-to-end encryption, optional metadata minimization, unlinkable fund delivery, and extensible privacy enhancements, the protocol provides strong protection against surveillance, leakage, and coercion. It enables secure coordination not only in a technical sense, but also in a social and economic context where privacy is a prerequisite for freedom and trust.

Implementation Roadmap

The rollout of PGNChain follows a phased implementation strategy that reflects the protocol's modular architecture, emphasizing practical utility, developer adoption, and incremental decentralization. Rather than attempting to deliver an all-encompassing system from the outset, PGNChain's roadmap prioritizes the rapid deployment of core messaging and value delivery features, followed by progressive enhancements in privacy, scalability, and integration tooling. Each phase is designed to deliver standalone value, allowing users and developers to interact with a minimal viable protocol that can evolve naturally as demand and adoption grow.

The initial phase focuses on the launch of the core encryption and messaging infrastructure. This includes the browser-based encryption engine, which enables users to compose and decrypt messages using wallet-native public keys, and the integration of decentralized storage layers like IPFS or Arweave for hosting encrypted payloads. Alongside this, the MessageLog smart contract is deployed on one or more EVM-compatible chains to enable optional message anchoring, timestamping, and verifiability. A simple, web-based DApp allows users to compose encrypted messages, store them off-chain, and optionally register the reference on-chain. This establishes the protocol's fundamental utility—secure, wallet-to-wallet messaging—with minimal friction and no dependency on third-party intermediaries.

In the second phase, attention shifts to the value transfer layer. The PigeonVault smart contract is introduced, allowing users to deposit claimable funds using hash commitments that can be shared via encrypted messages. The web interface is extended to support vault interactions, enabling users to create, monitor, and claim deposits based on secret proofs. This transforms PGNChain from a messaging tool into a communication-and-settlement protocol capable of transmitting both information and value. It also enables early use cases like stealth airdrops, crypto gifting, and DAO bounties, where privacy and coordination are equally essential.

Following the establishment of messaging and claim flows, the third phase involves the release of the PGNChain wallet extension or mobile-compatible app. This extension introduces background message scanning, push-style notification support, and inbox visualization. It provides a seamless user experience for encrypted messaging and vault claims, without requiring users to manually retrieve content identifiers or decrypt messages outside their wallet environment.

This stage also introduces meta-transaction capabilities and optional relayer infrastructure, allowing low-balance or gasless users to interact with the protocol via signature-based submission models.

The fourth phase focuses on privacy and scalability enhancements. Zero-knowledge proof systems are introduced for private claims, stealth vault interaction, and verifiable messaging without metadata exposure. This phase also considers integration with decentralized identity frameworks to enable selective disclosure and attestations, without relying on public key disclosure or centralized registries. Scalability improvements may include the introduction of protocol-specific relayer networks, message batching, and Layer 2–optimized deployment contracts to minimize gas costs and increase throughput.

The final phases of the roadmap center on developer adoption and ecosystem expansion. This includes the release of SDKs in multiple languages such as JavaScript, TypeScript, and Rust, allowing third-party applications, wallets, and DAOs to embed PGNChain functionality natively. Comprehensive documentation, message schemas, testnets, and code examples are delivered to support open development. At this point, governance structures for PGN token holders may be introduced to guide protocol evolution, approve integrations, and manage ecosystem incentives through proposals and on-chain voting.

Through this incremental and modular roadmap, PGNChain builds its foundation from secure communication toward full-spectrum coordination. Each layer adds composability and control, enabling the protocol to evolve organically into a cornerstone messaging and micro-settlement layer for Web3 infrastructure.

Conclusion

PGNChain introduces a vital missing primitive to the blockchain ecosystem: encrypted communication and claim-based value delivery that is composable, private, and verifiable. By decoupling message content from on-chain state and relying on wallet-native encryption, the protocol creates a trustless environment for exchanging information and tokens without exposing user identities, transaction intent, or relational metadata. It enables participants to coordinate directly, securely, and asynchronously—whether for DAO governance, decentralized compensation, stealth airdrops, or personal messaging—without relying on external platforms or centralized relays.

Unlike traditional attempts at on-chain messaging that burden chains with heavy data storage or rely on off-chain servers vulnerable to surveillance and censorship, PGNChain embraces minimalism and modularity. It uses off-chain encryption and decentralized storage to preserve confidentiality, while leveraging on-chain logs and smart contract vaults only when needed to enable verifiable claims and settlements. This architecture allows the protocol to operate across multiple chains, remain low-cost, and scale horizontally as part of any application stack.

Through the $PGN token, PGNChain also embeds an incentive mechanism that supports privacy, accessibility, and community-led growth. Whether paying for message anchoring, funding relayers, unlocking privacy features, or participating in governance, $PGN aligns the protocol's economic flows with its usage and adoption. As users interact with the protocol, they not only benefit from secure messaging and confidential transfers but also help maintain and evolve the network itself.

Ultimately, PGNChain is not a product but a protocol—an invisible infrastructure layer that makes trustless coordination possible between humans and machines. It creates the conditions for secure, composable communication to exist within the Web3 paradigm and expands the expressive power of blockchains beyond tokens and contracts into the realm of intent, negotiation, and private exchange. In doing so, it redefines what is possible for decentralized systems, enabling a future where communication and value can move together—securely, silently, and without compromise.