P-20 BlockChain
Native coin: Private USD (PVT-USD)

Indigo Shell

June 10, 2023

1 Introduction

Despite their widespread use, stablecoins like USDT (Tether) and USDC face significant drawbacks and challenges. Their centralized nature contradicts the decentralized principles of cryptocurrencies, raising concerns about transparency, independence, and potential manipulation. Reliance on fiat currencies also poses risks to trustworthiness and stability. Recent incidents, such as the "USDC" depegging, highlight the crucial significance of effective risk management in the cryptocurrency ecosystem, ensuring its stability and fostering trust among participants.

This white paper introduces PVT-USD, a revolutionary stablecoin leveraging the power of the P-20 Blockchain to provide enhanced privacy, security, and stability. Built upon advanced cryptographic principles, PVT-USD aims to address the limitations of existing stablecoins by combining decentralized issuance, and robust privacy features.

PVT-USD stands out with its commitment to privacy-preserving smart contracts, setting it apart from other stablecoins. Through the use of cryptographic techniques such as secure multi-party computation, PVT-USD ensures that the execution of smart contracts maintains user privacy. This advanced approach safeguards sensitive information and prevents any unauthorized disclosure during contract execution. By incorporating privacy-preserving smart contracts, PVT-USD enhances the overall privacy and security of the network, providing individuals with a trusted platform for conducting confidential and secure transactions.

2 USDT drawbacks and potential challenges

2.1 Traceability and Privacy Concerns

Privacy and anonymity are essential in electronic cash to conceal transactions from third-party observation. Two key properties for a fully anonymous electronic cash system are Untraceability and Unlinkability.

Untraceability: In a fully anonymous electronic cash system, the sender's identity remains untraceable, ensuring transaction privacy.

Unlinkability: In anonymous electronic cash, preserving recipient privacy involves preventing the linking of outgoing transactions to the same recipient.

USDT, USDC, and other stablecoins' lack of untraceability compromises anonymity and their centralized nature poses risks of fund freezing and potential collapse.

2.2 The Proof-of-reserve Controversy

Tether and other USD-backed stablecoins face doubts about their stability due to bank failures, bringing uncertainty in maintaining their peg, despite audits and transparency following stablecoin crashes like USDC.

2.3 Centralized Nature

Tether's admin control features have caused controversy by allowing transaction freezing and reversal, leading to concerns about censorship.

2.4 Scalability Challenges

Tether (USDT) struggles with scalability on Ethereum, causing slow processing and high fees during peak demand. Resolving scalability is vital for Tether's reliability and efficiency in the cryptocurrency market.

3 The Private USD (PVT- USD) Technology

P-20 Blockchain is a technologically advanced and highly efficient blockchain ecosystem that incorporates robust cryptographic techniques, consensus algorithms, and privacy-preserving smart contracts. With a remarkable transaction capacity of 300,000 TPS, it offers unparalleled speed and scalability.

Leveraging ring signature cryptography, P-20 Blockchain ensures transaction confidentiality, untraceability, and unlinkability, prioritizing user privacy and anonymity. This integration of advanced cryptographic protocols enhances network security, making P-20 Blockchain a trusted and tamper-resistant stablecoin solution with exceptional privacy features. It revolutionizes the stablecoin landscape, providing a secure and efficient platform for confidential transactions.

4 P-20 Blockchain Ring Signature Algorithm

Ring signatures and RingCT are essential cryptographic primitives employed in the P-20 blockchain ecosystem, powered by the native currency PVT-USD. These techniques enable anonymous transaction signing and confidential amount transfers, ensuring user privacy and data protection. By integrating ring signatures and RingCT, P-20 offers a secure framework for private and confidential transactions, upholding the principles of transparency and security while preserving sensitive information.

4.1 Ring Signature Algorithm

The Ring Signature Algorithm is a crucial component of the P-20 blockchain ecosystem, enabling users to sign transactions on behalf of a group without disclosing the specific signer. This algorithm follows a series of steps to generate a valid ring signature.

Key Generation: In the key generation phase, each participant in the ring generates their public-private key pair using elliptic curve

cryptography. The public keys serve as unique identifiers, while the private keys remain securely held by the respective users.

Signing Process: The ring signature algorithm creates a ring signature for a transaction by selecting a random signing parameter and calculating commitment values based on the transaction message, initiator's public key, and the signing parameter. Each participant adds their signature component derived from their private key and the commitments of previous signers. This iterative process yields the final ring signature, ensuring signature integrity, authenticity, and preserving signer privacy and anonymity.

The Ring Signature Algorithm leverages complex mathematical operations, such as elliptic curve point addition and scalar multiplication, to generate valid ring signatures while preserving the privacy and anonymity of the signers. By employing this algorithm, the P-20 blockchain ensures secure and private transactions, maintaining the integrity and confidentiality of participant identities.

4.2 Key Generation process

Here's a more detailed explanation of the Key Generation process in ring signatures:

1. Each user in the ring generates a public-private key pair (pk, sk). For each user i, let's denote their public key as pk_i and their private key as sk_i. Public key pk_i is a point on an elliptic curve, and private key sk_i is a scalar value.
2. Let's consider an elliptic curve group defined by a curve equation: E: $y^2 = x^3 + ax + b(mod\ p)$. Here, a and b are curve parameters, and p is a large prime number representing the field size.
3. Key Generation for each user i:
    - Choose a random integer d_i such that $1 \leq d_i < p$. This will be the private key for user i: $sk_i = d_i$.

- Compute the corresponding public key for user i as follows: Compute the base point on the elliptic curve: $G = (x_G, y_G)$. Calculate the public key point $pk_i = d_i * G$, where $*$ denotes scalar multiplication. Scalar multiplication is performed using elliptic curve point addition and doubling operations. For example, if $d_i = 5$, then $pk_i = G + G + G + G + G$.

4. The resulting public key pk_i is a point on the elliptic curve that can be represented as (x_i, y_i). This public key $pk_i = (x_i, y_i)$ is shared publicly, while the private key $sk_i = d_i$ is kept secret by user i.

The key generation process ensures that each user in the ring has their own unique public-private key pair, enabling them to participate in ring signatures and perform cryptographic operations securely.

4.3 Signing Algorithm

Here's a detailed explanation of the Signing Algorithm in ring signatures:

To sign a message M on behalf of a ring $A, B, C, ..., Z$ using a ring signature:

1. Select a random number r as the signing parameter for the ring signature.

2. Compute a commitment C based on the message M, the public key pk_A of the user A in the ring, and the signing parameter r : $C = H(M, pk_A, r)$, where H is a hash function that takes the message M, the public key pk_A, and the signing parameter r as inputs and produces a commitment value C.

3. Compute the signature component s_A for user A as follows: $s_A = r - sk_A * Cmodq$, where sk_A is the private key of user A, and q is a large prime number.

4. Select a random user B from the ring other than A and send the commitment C and the signature component s_A to B.

5. When user B receives (C, s_A), perform the following steps:
   - Select a random number r_B as the signing parameter for user B.
   - Compute a commitment C_B based on the message M, the public key pk_B of user B, and the signing parameter r_B : $C_B = H(M, pk_B, r_B)$.
   - Compute the signature component s_B for user B as follows: $s_B = r_B - sk_B * C_B mod q$, where sk_B is the private key of user B.
6. Select another random user C from the ring other than A and B and send the commitment C_B and the signature component s_B to C.
7. Repeat step 5 for each user in the ring, passing the commitment and the signature component to the next user until the last user Z is reached.
8. When the last user Z receives (C_Z, s_Z), perform the following steps: Compute the final signature component s_Z for user Z as follows: $s_Z = r_Z - sk_Z * C_Z mod q$, where sk_Z is the private key of user Z.
9. User A combines all the received signature components $(s_A, s_B, ..., s_Z)$ and the corresponding commitments $(C, C_B, ..., C_Z)$ to produce the final ring signature.

By following these steps, a ring signature is created, allowing the message M to be signed on behalf of a group of users without revealing which specific user produced the signature. The use of random signing parameters and commitments ensures the unlinkability and privacy of the signer within the ring.

4.4 Verification Algorithm

Here's a detailed explanation of the Verification Algorithm in ring signatures:

To verify the authenticity of a ring signature $(C, s_A, s_B, ..., s_Z)$ on a message M:

1. For each user i in the ring $A, B, C, ..., Z$, perform the following steps:

   - Retrieve the public key pk_i of user i.
   - Retrieve the commitment C_i and the signature component s_i corresponding to user i from the ring signature.

2. Verify the ring signature for each user i by checking if the following equation holds: $s_i * G = r_i * G + C_i * pk_i$, where G is the generator of the elliptic curve group.

3. If the verification equation holds for all users i in the ring, the ring signature is valid, and the authenticity of the message M is confirmed.

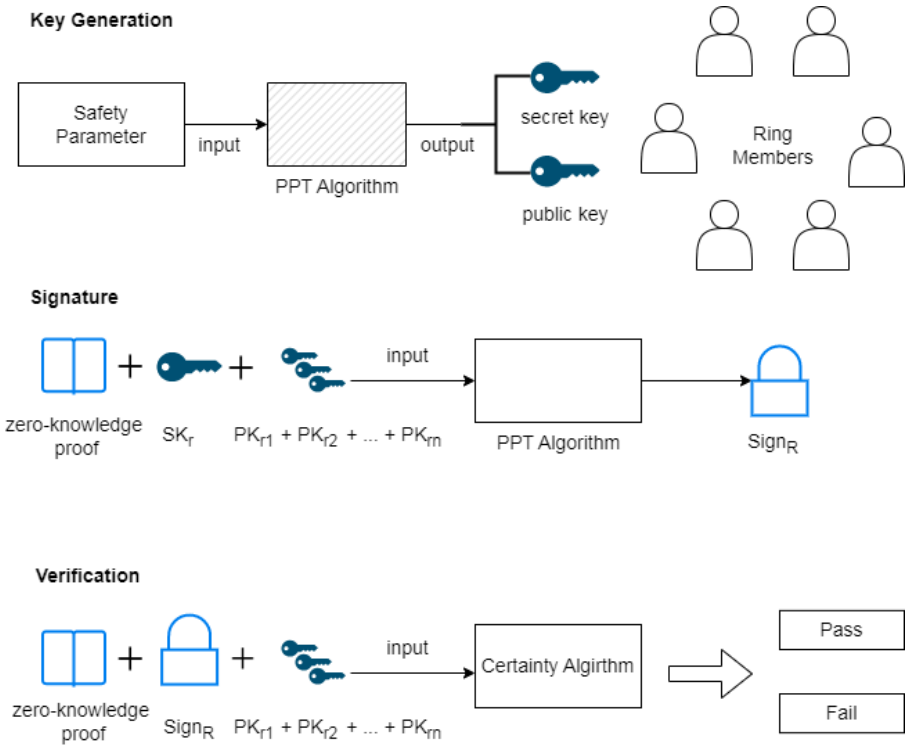

Figure 1: Ring Signature Algorithm

By performing the verification algorithm, each user in the ring independently verifies the ring signature components and ensures that they

satisfy the verification equation. If the equation holds true for all users, it indicates that the ring signature is valid and the message was signed by a legitimate participant in the ring.

The verification algorithm allows anyone to verify the authenticity of the ring signature without knowledge of the private keys of the ring members. It provides a way to ensure the integrity and validity of the ring signature scheme.

4.5 Ring Confidential Transactions (RingCT)

Ring Confidential Transactions (RingCT) is a cryptographic technique that enhances privacy in cryptocurrency transactions by hiding the transaction amount while ensuring transaction integrity. It extends the functionality of ring signatures to incorporate confidential handling of transaction amounts.

When Jeff wants to send a transaction to Rosie using RingCT, the following steps are involved:

1. **Transaction Creation:** Jeff creates a transaction with specified inputs and outputs.
2. **Ring Selection:** Jeff selects a ring of users, including Rosie, from the blockchain, ensuring Rosie's public key is among the keys in the ring.
3. **Ring Signature Generation:** Jeff generates a ring signature using the selected ring of users and transaction details, following the principles of the ring signature scheme. This signature guarantees the authenticity of the transaction while concealing the identity of the specific signer.
4. **Amount Confidentiality:** Jeff encrypts the transaction amount using a commitment scheme like Pedersen commitments. The encrypted amount, denoted as EncryptedAmount, ensures the confidentiality of the transaction value.

5. **Transaction Construction:** Jeff constructs the transaction with the ring signature, Pedersen commitments for the transaction inputs and outputs, and the EncryptedAmount.

6. **Transaction Broadcasting:** Jeff broadcasts the transaction to the network, making it available for verification and inclusion in the blockchain.

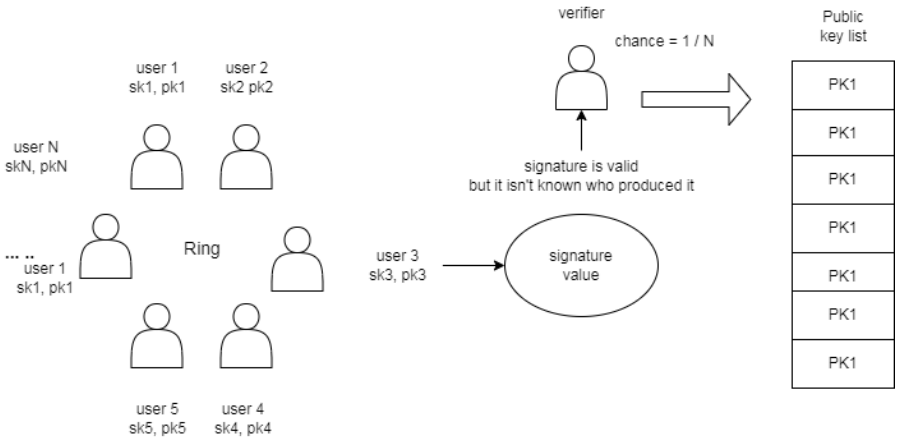

Figure 2: Ring Confidential Transaction (Ring CT))

Upon receiving the transaction, Rosie can perform the following steps to validate and process it:

1. **Ring Signature Verification:** Rosie verifies the ring signature to ensure the transaction's authenticity and integrity.

2. **Amount Decryption:** Rosie decrypts the EncryptedAmount using her private key, revealing the transaction amount. She can then update her balance accordingly based on this information.

By utilizing RingCT, Jeff and Rosie can conduct transactions privately, preserving their financial privacy while upholding the security and integrity of the P-20 blockchain network.

5 Consensus Mechanism

P-20 block-chain with its native currency Private-USD (PVT-USD) working on the below consensus network utilizes directed acyclic graphs (DAGs) to establish transaction order and validity consensus. Events, representing transactions, are organized within the DAG and assigned unique Lamport timestamps for logical ordering. The algorithm considers event dependencies, Lamport timestamps, and employs voting mechanisms to determine event ordering. Nodes participate in voting and virtual voting, expressing opinions on event popularity. Popularity scores, calculated from received votes, help estimate the consensus view. The resulting consensus order forms the network's transaction history, ensuring reliable and agreed-upon sequencing of events.

In addition to event ordering, our consensus algorithm incorporates asynchronous Byzantine fault tolerance (ABFT) principles. ABFT ensures the system's resilience against faults and malicious behavior within an asynchronous network environment. By combining event ordering, voting mechanisms, and ABFT, the algorithm provides a robust and secure consensus protocol that enables decentralized networks to achieve reliable transaction processing and consensus. This protocol ensures the integrity and accuracy of transactions within the PVT-USD network, fostering trust and reliability among network participants.

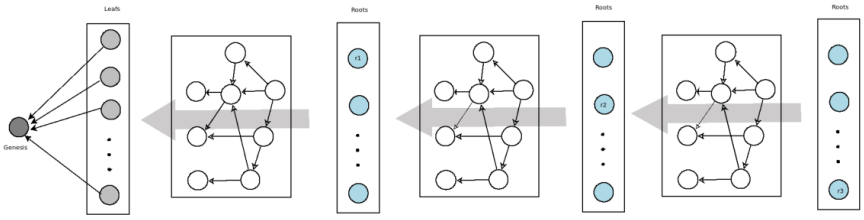

Figure 3: Consensus Method through Path Search in a DAG (combines chain with consensus process of pBFT)

Let's consider the consensus algorithm in the context of actual transactions:

5.1 Event Ordering

Each event in our consensus algorithm represents a transaction in the Private-USD (PVT-USD) network. Transactions are assigned unique Lamport timestamps, denoted by $L(t)$, where $'t'$ represents the transaction's logical timestamp. The Lamport timestamps allow us to order the transactions based on their occurrence in the system.

5.2 Asynchronous Byzantine Fault Tolerance (ABFT)

The algorithm incorporates ABFT principles to ensure that transactions are processed securely, even in the presence of Byzantine faults. ABFT models leverage mathematical concepts like formal logic, probability theory, and game theory to analyze and design fault-tolerant systems.

5.3 Event Flow

Transactions flow through the network as events. Each node in the network generates events by processing incoming transactions and creating corresponding event objects. When a node receives a new transaction, it validates the transaction's integrity, authenticity, and other properties to ensure its validity.

5.4 Event Ordering and Causality

Events (transactions) in the algorithm form a directed acyclic graph (DAG) based on causality. If transaction A references transaction B, it means that B causally precedes A in the DAG, indicating that B must be processed before A.

5.5 DAG-Based Voting

Nodes in the network engage in voting to determine the popularity of transactions and establish consensus on their order. Each node collects votes from other nodes regarding the popularity of events (transactions) they have observed. Votes are aggregated to determine the popularity score of each transaction, indicating the level of agreement among the nodes.

5.6 Virtual Voting

Virtual voting allows nodes to vote on transactions they have not directly observed. Nodes calculate the popularity of a transaction based on the votes received from other nodes in the network. The popularity score reflects the collective opinion of the network about the order of transactions.

5.7 Determining the Consensus Order

Based on the voting results and popularity scores, nodes determine the consensus order of transactions. Nodes assign timestamps to transactions based on the Lamport timestamps and the voting outcome. The consensus order represents a total order of transactions agreed upon by the network.

In the context of actual transactions, the consensus algorithm ensures that transactions are ordered, validated, and processed securely, even in an asynchronous and potentially Byzantine network environment. The mathematical concepts and algorithms embedded within the protocol help achieve fast and reliable transaction confirmation.

6 Price Stability

Let's delve into a more detailed mathematical explanation of Private-USD (PVT-USD)'s strategy, including the rebase mechanism and the target price adjustment.

6.1 Rebase Mechanism

Our rebase mechanism adjusts the supply of PVT-USD coins held by each user proportionally, aiming to bring the price back to the target. Here's a step-by-step breakdown of the mathematical calculations involved:

- Let's denote the total supply of PVT-USD coins at time t is $S(t)$.
- The price of 1 PVT-USD coin at time t is denoted as $P(t)$.

- The scaling factor for the rebase at time t, denoted as $R(t)$, is calculated as:

$$R(t) = S(t)/P(t) - 1$$

The scaling factor represents the percentage change in supply during the rebase.

- Each user's balance is adjusted based on the scaling factor. Suppose a user holds x PVT-USD coins before the rebase. After the rebase, their post-rebase balance, denoted as x', is given by:

$$x' = x * (1 + R(t))$$

If $R(t)$ is positive, indicating that the price $P(t)$ is higher than the target price, users' balances increase, resulting in a supply expansion. Conversely, if $R(t)$ is negative, indicating that the price $P(t)$ is lower than the target price, users' balances decrease, leading to a supply contraction.

6.2 Target Price Adjustment

The algorithm adjusts the target price, denoted as P_0, over time to maintain stability. The target price is influenced by the deviation of the actual price $P(t)$ from P_0. Here's the mathematical explanation:

- The adjustment factor, denoted as α, is calculated based on the deviation of the price $P(t)$ from the target price P_0:

$$\alpha = (P(t) - P_0)/P_0$$

The adjustment factor α represents the percentage change required to bring the price $P(t)$ back to the target price P_0.

- The updated target price, denoted as P_0', is determined by adjusting the previous target price P_0 using the adjustment factor α:

$$P_0' = P_0 * (1 + \alpha)$$

- This adjustment helps guide the target price toward stability. When the price $P(t)$ deviates from P_0, α becomes positive or negative, influencing the target price to move closer to the actual price.

P-20 Blockchain maintains price stability through dynamic supply adjustments and target price adaptation, mitigating volatility for a reliable native currency value.

7 Privacy-Preserving Smart Contracts on the P-20 Blockchain

The P-20 blockchain employs privacy-preserving smart contracts, advanced cryptography, and secure computing environments to address privacy concerns. It ensures confidentiality, supports complex computations, and maintains data integrity. Key approaches used in the P-20 blockchain for privacy-preserving smart contracts include:

1. **Trusted Execution Environments (TEE):** The P-20 blockchain utilizes trusted execution environments to enable secure and trustless execution of smart contracts. This ensures data confidentiality, scalability, and inaccessible program states to external entities.

2. **Secure Multi-party Computation (sMPC):** The P-20 blockchain utilizes secure multi-party computation to ensure privacy during computations on distributed data. It divides data into pieces, conducts operations, and combines results without exposing the original data, providing robust privacy guarantees.

3. **Zero-Knowledge Proofs (ZKP):** Zero-knowledge proofs integrated into the P-20 blockchain enable privacy-preserving verification of computation correctness, allowing nodes to create verifiable proofs while maintaining privacy and enabling public validation.

In summary, the P-20 blockchain utilizes trusted execution environments, secure multi-party computation, and zero-knowledge proofs to enable privacy-preserving smart contracts and secure data handling, ensuring confidentiality and data protection.

8 Conclusion

In conclusion, PVT-USD represents a significant advancement in the realm of stablecoins. By leveraging the P-20 Blockchain and incorporating privacy-preserving smart contracts, it introduces a novel approach to addressing the limitations of existing stablecoin solutions. PVT-USD ensures the confidentiality of user transactions through its implementation of ring signature cryptography, offering untraceability and unlinkability for enhanced privacy and anonymity.

The integration of advanced cryptographic techniques not only enhances user privacy but also strengthens the overall security and trustworthiness of the network. PVT-USD aims to establish itself as a decentralized stablecoin solution that aligns with the principles of cryptocurrencies while providing robust privacy features. By combining decentralization, privacy, and security, PVT-USD offers a reliable and trusted platform for conducting confidential and secure transactions, revolutionizing the stablecoin landscape and providing users with an efficient and privacy-focused alternative.