canopus

Decentralized Galactic Network | Avalanche Ecosystem

Canopus is the first decentralized galactic network protocol in Avalanche Ecosystem which is the main focus on stability, improvement transaction speed and reducing the fee.

Canopus (OPUS) is unique functionality and control model never seen before in the Network Protocol. The platform also focuses on token issuance, synthetic staking between chains, interoperability, and more interesting cross-chain features in development.

We take all good parts from available coins/tokens and combined them with the improvement into Canopus (OPUS) Protocol.

Disclosure

The information described in this document is preliminary and is subject to change at any time. In addition, this document may contain forward-looking statements. Forward-looking statements generally refer to future events or our future results. This includes, but is not limited to, Canopus projected performance; expected
development of their business and projects; implementation of your vision and growth strategy; and the completion of projects that are currently underway, in development or otherwise under consideration. Forward-looking statements reflect the beliefs and assumptions of our management only as of the date of this presentation. These statements are not guarantees of future results and should not be relied upon too much. Such forward-looking statements necessarily involve known and unknown risks that may lead to actual and future results.
periods to be materially different from any forecasts expressed or implied herein. Canopus assumes no obligation to update the forward-looking statements. While forward-looking statements are our best projections at the time of their creation, there is no guarantee that they will be accurate as actual results and future events could differ materially. The reader is cautioned not to place undue reliance on forward-looking statements.

Introduction

Here are the main takeaways from the development of the OPUS Galactic model:

- The resources spent by the validator on staking are proportional to the total stake of the validator.
- The rewards accumulated by the validator for the validation are proportional to the total stake of the validator.
- Since there is no leader in Avalanche, there are no additional rich-becomes-rich effects.
- Validators who block their bet for a longer period receive a higher reward.
- Validators are encouraged to stay online and operate properly as their remuneration is based on uptime and validation.
- OPUS is a limited supply token with a maximum value of 1 billion tokens.
- Despite the limitation, OPUS is still manageable. The speed at which the maximum cap is reached depends on corporate governance.
- Commissions are not paid to a specific validator. Instead, they are burned, which increases the OPUS deficit.

Abstract

The rapid growth of data networks in recent years has led to a huge development in e-commerce. Internet banking and commerce are two important applications that process financial transactions from all over the world. This allows banks and merchants to streamline financial transaction processes and provide customer-friendly 24-hour service. On the one hand, labor and infrastructure costs have plummeted, while shipping costs, third-party copyright and customer information protection costs rise. Electronic micropayments are one of the most important research topics in e-commerce, especially in low-cost online and offline payment scenarios in rural areas. In this article, we will discuss some important micropayment plans, explore their advantages and limitations, and then suggest an improved micropayment plan. We will discuss two main micropayment plans using the public / private key concept, and then consider another plan that uses the hash chain concept. In this and related schemes, we observe certain limitations that induce us to expand one of the attractive plans, namely the plan of Hwang and Sung [8], towards more efficient, protecting all other functions without compromising the reliability of the plan. We compare the improved design with others and show that
the improved design provides better safety and efficiency; This ensures that the scheme is valid for real-world applications, especially in resource-constrained environments such as mobile payments using handheld devices or a customer's chip card. debit / credit transaction through the POS terminal._____

Assumption

Due to the growing importance of intangible goods in the global economy and their immediate delivery at low cost, traditional payment methods tend to be more expensive than the actual product. Digital content or online services can be less expensive than the smallest currency in the physical world. There are two ways to manage electronic payments: online payments, where the merchant verifies the payment sent by the buyer with the broker before serving the buyer, and double spend, offline payments to avoid excessive costs, so the online connection with the broker does not work. mandatory.

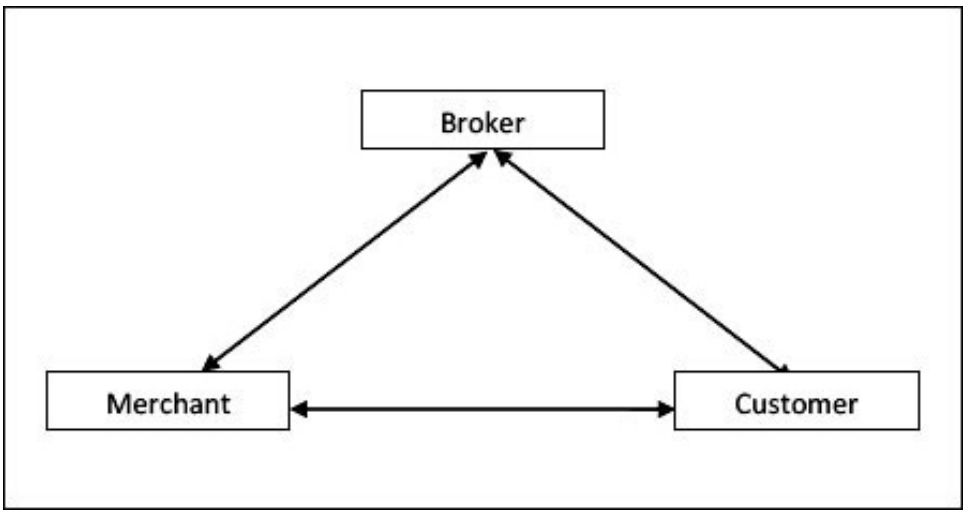
Electronic payment models [2], [13], [16] can also be divided into two groups in terms of network requirements, stability of the consumer base and main payment methods. The first is a transaction-by-transaction model in which individual payments do not require prior agreement between buyer and seller. The other is payments through accounts with a broker before the actual payment transactions of the buyer and seller are processed, and there must be some sort of agreement between them. Transactional payments can be divided into two subgroups: Credit card payment transactions are usually designed for payments with high fees, such as a few tenths, hundreds, or even thousands of dollars. In contrast, clean money transactions are usually small-value payments with high transaction costs and online requirements, so they are very similar to the idea of micropayments. The downside to processing credit card payments is the cost of transactions to the clearinghouse they contact, especially for merchants who have to pay certain bills. Of course, this will directly affect the pricing policy and the interest of potential customers. A special type of electronic payment system called micropayments (Site 2), which deals with small transactions of digital content and services, depending on the amount of the transaction, increases the interest of the research community. Some important applications of micropayments are electronic publishing, multimedia services, games, etc. Applications, due to the small amount of the transaction, the trader receives a relatively financial profit from each transaction. As a result, hardware, software, network, etc. Costly computations such as digital signatures and databases should be limited to reduce investment. Recently, micropayments [3], [4], [6] - [10], [12], [14], [15], [17], [18], [21] - [26]] offer income in the Internet. a relatively significant innovation in the stream. The core of micropayments is to take advantage of a large audience volume by offering content at a low cost. Other variations on this idea are that fractional parts are charged for the same fractional amount of content, for example, one cent for one web page of an online magazine. Small payment amounts and high frequency of transactions in the e-commerce network are important factors in a micropayment system.

1. Requirements for Micro-payment System

A micro-payment system comprises the following entities [16]:

- Client. The client buys e-money / e-money from brokers using real money through a macro payment system. The customer can then use e-money / e-money to make purchases using micropayments.
- Seller: The seller is a "data bank". Provides customers with data, services, or both.
- Broker. The broker acts as an intermediary between traders and clients in order to simplify the tasks they perform. It usually acts like a bank and provides e-currency for micropayments. The general scheme of the micropayment system operation is shown in Figure 1.
- Anonymity: e-money should not provide information to any user, in other words, it should be an anonymous e-money transaction.
- Divisibility. Since it is the basic unit of currency, electronic currency can be split up or presented with changes.
- Transfer: Electronic currency can be transferred to a third party by offering the corresponding amount of money.
- Prevention of double spending: e-money cannot be used a second time unless it is counterfeit.

Figure 1: Micro-payment System



Electronic payments are stored and transmitted digitally, which creates new challenges for the development of secure electronic payment systems: payments are easy to copy, unlike traditional physical payment instruments. Since digital payments are represented as simple byte strings or bit strings, nothing in them prevents them from being copied in this way. If the security of a payment system depends on how payments are hidden from outsiders, anyone with access to payments can probably use them multiple times. We can see that anonymous cash transactions are an important requirement, and at the same time efficiency is another issue. In this article, we discuss a merchant-specific micropayment system [18]; A large number of merchant-supported micropayment systems [10] followed by a scheme that provides anonymity using the concept of blind signature and mixed chain [8]. Next, we present an enhancement to the hidden signature phase used in schema [8] to achieve greater efficiency without compromising security.

The rest of the article is organized as follows: Section 2 discusses three micropayment plans [18], [10], [8], which significantly influence the creation of a new micropayment plan. The first two schemes [18], [10] are considered as basic micropayment systems. Third, [8] is an efficient option that supports multiple vendors and anonymity features. Chapter 3 introduces improvements to the plan [8]. Chapter 4 analyzes the improved schema. We conclude the article with Chapter 5.

2 Related Work

The Canopus system (Site 1), [3], [4] is based on a one-time token system. The user creates electronic banknotes with blunt edges and sends them to their bank for signature with their bank's public key. Bank accounts are signed, debited from the user's account, and the signed accounts are sent back to the user. The user eliminates the glare factor and uses m to buy the store. The store verifies the authenticity of the banknotes using the bank's corresponding public key and sends them to the bank, where they were verified against a previously used list of banknotes. The amount is credited to the store's account, the deposit is confirmed, and then the store ships the item. Millicent [7] is a decentralized micropayment scheme that allows payments from 1 / 10th of a cent. It uses an electronic currency called "Script". It is designed to make the cost of the fraud greater than the cost of the actual transaction. It uses symmetric encryption for all data transactions. Millicent is a lightweight and secure Internet e-commerce protocol. It is designed to support purchases worth less than a cent. It is based on decentralized verification of electronic money on the merchant's server without additional communications, expensive encryption or offline processing. Then, coupon plans, lottery tickets, and coin-tossing protocols [6], [12], [17] were proposed that can minimize the number of messages included in each transaction. Electronic coupons are transferable, but no direct exchange between buyers and sellers is possible. As a result, a financial intermediary is needed, and this will increase the cost of swap transactions. In addition, the draw and lottery protocols are based on the assumption that economic agents are not exposed to risk and will be satisfied with fair rates. Over the years, several new plans and improvements for micropayments have been proposed [1], [8] - [10], [14], [15], [18], [21] - [26]. In this section, we will look at two basic plans [10], [18] for a general understanding of the micropayment system, and then discuss a more efficient and flexible plan [8]. Interested readers can refer to [16] for some other schemes and a wide range of functions and design criteria for a micropayment system.

We use the notation shown in Table 1 when discussing the following chapters of the article.

U	Customer
M	Merchant
B	Broker
ID_X	Identity of X, where X ∈ {U, M, B}
A_X	Address of X, where X ∈ {U, M, B}
C_U	U's certificate
E_U	U's certificate expiry information
E_M	Expiry information for redemption
I_U	U's certificate serial number credit unit info.
OI	Order information (category, amount, etc)
PK_X	Public key of X, where X ∈ {U, M, B}
SK_X	Private key of X, where X ∈ {U, M, B}
r_X	Random number chosen by X, where X ∈ {U, M, B}
P	A generator point on elliptic curve
h( )	Cryptographically secure hash function
K	Secret key of B
{m}K	M is encrypted under key K
⊕	Exclusive-OR operator
||	Concatenation operator

Table 1: Notation used in the Schemes

2.1 The Payword Scheme

In 1996, Rivest and Shamir [18] proposed the PayWord system, which is a credit-based system. The system uses RSA public key cryptography [19] and the concept of hash chain [20]. In the PayWord system, when a registered customer asks the merchant for a service, s/he needs to create a PayWord chain that acts as the currency made payable to the merchant. The merchant then needs to inspect whether the customer is legitímate and the paywords (coins of the PayWord system) chain is generated by the customer.
Later, the merchant collects customer's paywords and redeems the payment from the broker. The PayWord system reduces the number of on-line communications between broker and merchant because the merchant does not need to settie for every purchase. The paywords chain generation with length n can be expressed as wi = h(wi+i), where i = n-1, n-2, ..., 0. When generating the paywords, the customer selects a random number wn called a seed. Then wn is hashed iteratively in reverse order until the root of the chain w0 is generated. During shopping, the customer in order from w1 to wn releases paywords, and the merchant verifies it easily by hash operation.
PayWord is a merchant-specific payment system, that is, the paywords chain is spent only to a particular merchant. When a customer contacts a new merchant M for requesting service, besides generating a new chain, the customer must send a commitment to M. The commitment contains the identity of the merchant, the certificate issued by the broker, the root of an unused chain, the current date and other information.
To execute continuous transactions, the customer pays paywords within the chain which belongs to the merchant in order. After an appropriate period, the merchant will contact with the broker to request redemption. For each chain, the merchant sends the latest paywords s/he received and the customer's commitment to the broker; therefore, hashing from latest paywords to the root of the chain can valídate the correctness of transactions. If the validating is correct, the broker debits customer's account with the used length of the chain and credits merchant's account with the same amount.

We depict the transaction process of the Payword system in tfigure 2.

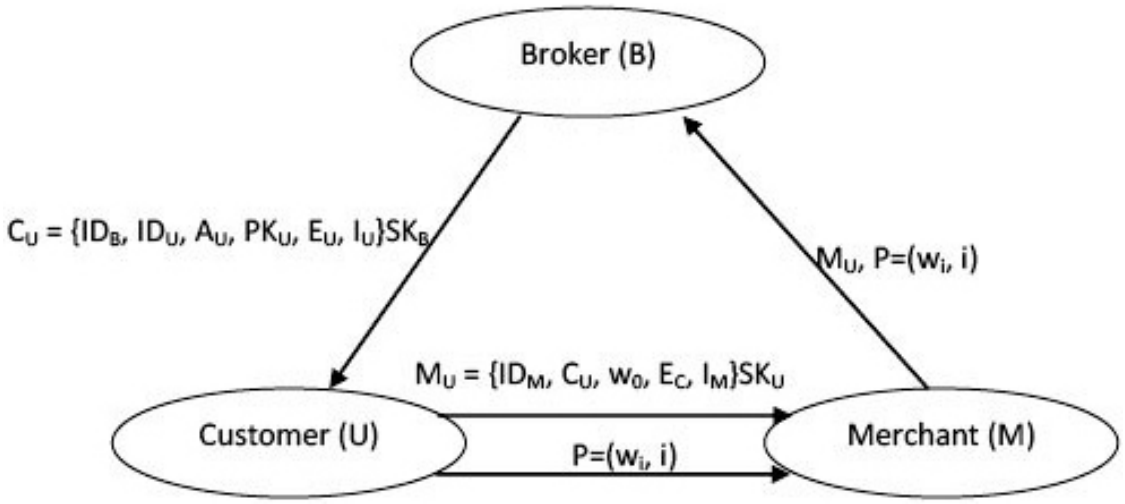


Figure 2: The Canopus Scheme

Our Observations

PayWord is designed as a credit-based system. It takes advantage of hash chain to ensure computational efficiency, and achieves non-repudiation for each payment belonging to the same chain by only one signature. After receiving a certificate , a customer is authorized to transact with a merchant in a specified amount without the oniine communication to broker, which reduces the communication and the risk of the broker becoming a bottleneck, provides the customer more flexibility. However, the system suffers from the following limitation:

- PayWord is a merchant-specific payment system; therefore, customers have to maintain tupies of particular data of chains corresponding to distinct merchants;
- Customer has to perform hash chain operations as many as the number of merchants each time s/he wants to transact business with;
- Customer has to store all the different paywords of each merchant and the last index used for the transactions; and
- Customer could make payments exceeding her/his authorized credit limit. 2.2 Kim and Lee's scheme

2.2 Kim and Lee's scheme

In 2003, Kim and Lee [10] proposed a micro-payment scheme that supports multiple merchants. The scheme is divided into three phases: certificate issuing phase, payment phase, and redemption phase.

Certificate Issuing Phase: Customer requests a certificate to a broker by transmitting her/his private information via a pre-established secure channel. The broker sends Cu, which guarantees to be justified and Us which will be used for the root value in payment phase later. Each customer generates her/his public and private key pair (PKu, SKu) and sends PKu with lu that includes the maximum number of merchants N, the length of hash chain n as well as her/his credit card data to the broker. Since a customer's certificate signed by a broker, those who intend to use this key must trust him.
The broker creates special data Tu, which acts as a key factor of the root value. It is used to make clear that the new hash values generated by the broker are issued to whom, since no one except the broker can create it.
Tu = h(U, rB , K), where K is the secret key of the broker
Su = {si, / si, = h(si+i, Tu), i = N-1, ..., 0,} where si, is generated by a shared user-broker secret key.
The certificate Cu, in which all the elements including the expiry date of the certificate Eu are signed by the broker and sent to the customer with Su and a nonce ru.
Cu = {IDB, IDu, PKu, Tu, lu, Eu}SKB

We depict the transaction process of Kim and Lee's protocol in tfigure 3.

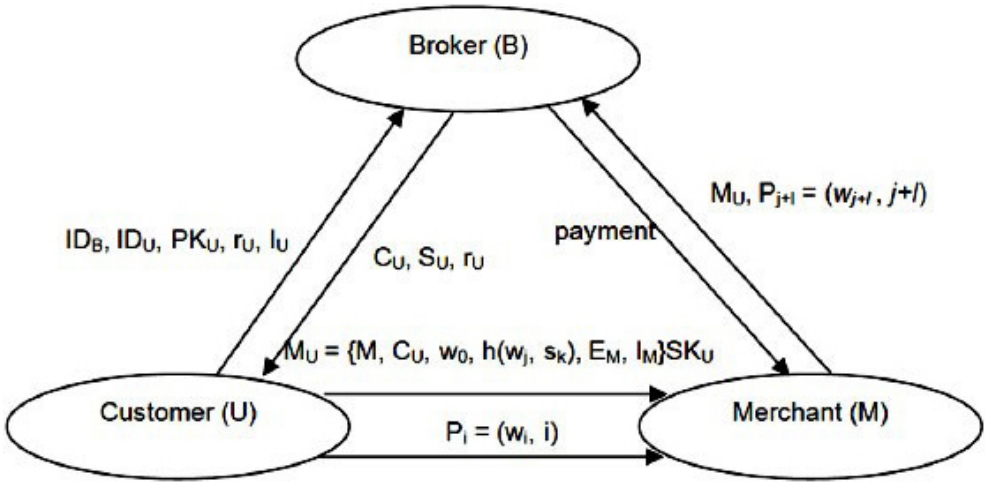


Figure 3: Kim and lee's Scheme

Payment Phase

Payment Phase: The root value of paywords is combined with si received from a broker, which enables a customer to use the rest of the unspent paywords in succession for multiple payments to different merchants. The customer who gets the certificate in previous phase can now create paywords and commitment. The commitment consists of the identity of the merchant with whom a customer intends to do business, the certificate , the root components which are modified into wj h(wj Sk), the expiry date of the commitment (EM), and other information (IM), where 0 ≤ j ≤ n used to setup root value for different merchants. Then the customer signs the components

$$Mu = \{V, Cu, w0, h(wj, sk), EM, IM\}SKU$$

In order to spend the rest of the paywords in succession, a customer should set the root value of paywords to be spent in next payment phase into the combination of the hash chain values respectively generated by a customer and a broker. For example, if it is assumed that a customer used paywords as many as Wj-1 in previous transactions and spent / paywords at the current transaction with kth merchant, the root value of paywords should be identical with h(Wj, Sk) to be proper for the payments. The customer can apply his paywords to different merchants up to the máximum transaction limit of N unless the last payword exceeds wN. The merchant stores the last received payment information of Pi+l= (Wj+l, j+l) and the commitment, and terminates the payment phase.

Redemption Phase: Merchant should carry out the redemption process with a broker within a preagreed period of time. The broker confirms whether the payment request of the merchant is correct or not by verifying the certificate . The merchant first requests for redemption to a broker by sending a customer's commitment and payment information. From these data, the broker verifies his signature marked at the certificate and redeems Pi+l to a corresponding amount of money. We note that a broker can verify paywords only from wj to Wj+l for that request. In other words, since the corresponding root value is wj, the only thing imposed to the broker is that the last received payword wj+l is identical with wj by applying hash function / times. The broker processes redemption requests from merchants less than N before being overdue. Finally, the broker completes the redemption process if the last received value wlis less than the máximum value of the hash chains.

Our Observations: The scheme supports multiple merchant payments and prevenís overspending payment. Moreover, in PayWord system, whenever a customer wants to establish transactions with each vendor, s/he has to obtain a certificate from a broker and create a series of paywords, while a customer is able to make transactions with different merchants by performing only one hash chain operation in Kim and Lee's scheme. Nevertheless, we observe the following limitation on this scheme:

the system performance is reduced by necessarily frequent signing in each transaction;
the customer has to keep different hash chains and corresponding Indices, however the overhead of merchants is relatively high. To securely deposit,
the bank has to collect all Canopus belonging to the same chain. It needs an additional storage space and wastes undetermined waiting time; and
a dispute arises if the merchant forges transaction records or the customer double spends.

2.3 Hwang and Sung's scheme

Hwang and Sung [8] proposed a micro-payment scheme using hash chain [20] that provides customer's anonymity by applying a blinding signature using elliptic curve cryptography [11]. As we propose an improved version of this scheme in next section, which provides more efficiency than its current performance, we explain a bit more on this scheme so that a straightforward comparison can be established while measuring performance of the schemes in section 4.2. The scheme is divided into four phases: registration phase, blinding phase, transaction phase, and redemption phase.
Registration Phase: Both customer and merchant have to register with a broker. The customer shares a secret key KUB with the broker, and the merchant shares a secret key KMB with the broker. The customer selects a pseudonymous identity IDU, which is unique to every customer. The broker selects a master secret key k and keeps it secret from others.

Blinding Phase

The customer sends a withdrawal request to the broker before s/he requests any service from merchant. The phase works as follows:

Transaction Phase: U asks service from M by the following steps of operation:

Redemption Phase: M would carry out the redemption process with B by the following steps.

Our observations: The scheme supports multiple merchant payments and prevent overspending payment. Further, the scheme provides an important property, namely, anonymous cash transaction, which is strongly related to the practical applications in electronic business. However, to achieving anonymity property, the scheme uses a binding phase using elliptic curve cryptography, which increases overall cost of the scheme. This motivates us to work further for an efficient binding phase which provides both efficiency and security retaining all other features of Hwang and Sung's scheme intact.

Step 1: U sends $\{ID_U, I_U\}$ to B. After checking ID_U, B computes $R' = k \cdot P$ and sends R' to U.

Step 2: Upon receiving R', U selects a random number w_n and creates a hash chain
$w_n, w_{n-1}, \ldots, w_1, w_0$, where $w_i = h(w_{i+1})$, $i = n-1, n-2, \ldots, 1, 0$, and n is the limited amount that B allows U to spend.
Then U computes $R = uR' + vP$, where P is a generator point on elliptic curve
$m = h(R \,||\, w_0)$
$m' = m/u$,
where u and v are two secret random numbers and sends $\{m', n\}K_{UB}$ to broker.

Step 3: If n is smaller than the limited amount then broker calculates $S' = SK_B \cdot m' + k$ and sends it to the customer. Otherwise, B rejects U's request.

Step 4: Upon receiving S', U calculates $S = S'u + v$ and checks whether $S \cdot P = m \cdot PK_B + R$
If it does hold, s/he obtains a valid signature (R, S) on message m.

Step 5: Broker now creates two factors: T_U and S_U, where
$T_U = h(ID_U, r_B)$
$S_U = \{s_i \mid s_i = h(s_i + 1, T_U), i = N-1, \ldots, 0\}$, where N is the maximum number of merchants that a customer can do her/his business.

Transaction Phase: U asks service from M by the following steps of operation:

Step 1: U sends transaction request $\{A_M, ID_U, ID_B\}K_{UB}$ to B.

Step 2: B maintains a table of each U's ID_U and knows the secret key K_{UB}. Now B decrypts the request and checks U's authenticity. If U is authenticated, B creates a one-time session key K_{UM} for U and M, and sends $\{K_{UM}\}K_{UB}$ to U.

Step 3: Upon receiving K_{UM}, U calculates $R_{UM} = h(w_j \oplus (s_k \,||\, K_{UM}))$ and sends $\{R_{UM}, (R, S, m), w_0, (w_j, t), s_k, OI, Exp\}K_{UM}$ to M, where $t = j - i + 1$, and Exp denotes the date after which the hash chain is invalid. The Exp can limit the length of time and both merchant and broker need to store information about the state of a hash chain.

Step 4: M verifies the signature and R_{UM} as follows:
$S \cdot P = m \cdot PK_B + R$
$w_{n-1} = h(w_n)$, where $n = j-1, j-2, t, 1$
$h^{t-1}(w_j) = h^{t-2}(w_{j-1}) = \ldots = h(w_{j-t+2}) = w_{j-t+1}$
$R'_{UM} = h(w_{j-t+1} \oplus (s_k \,||\, K_{UM}))$
If these checks hold, M will sell electronic items or services to U.

3 An Improvement of the Hwang and Sung's scheme

We present a binding phase using the RSA-based binding signature [3], [5]. We show that this refinement makes Hwang and Sung's scheme more efficient, retaining all other features intact.
Blinding Phase: The customer sends a withdrawal request to the broker before s/he requests any service from merchant. The phase works as follows:

Step 1: Broker
1.1. Choose two large primes p and q, which are secret
1.2. Compute $\lambda = pq$
$\varphi(\lambda) = (p-1)(q-1)$
1.3. Select public key e such that $1<e<\varphi(\lambda)$ and $\gcd(e, \varphi(\lambda)) = 1$
1.4. Compute private key d satisfying
$ed \equiv 1 \bmod \varphi(\lambda)$

Step 2: Customer
2.1. Choose random numbers r and u
2.2. Compute $\alpha = r^e h(w_0)(u^2+1) \bmod \lambda$
2.3. Send (a, α) to the broker
Here the information 'a' can represent the expiry date, the amount of cash (upper limit) that the customer can use, the value of each hash coin.

Step 3: Broker
3.1. Choose a random factor $x < \lambda$
3.2. Send x to the customer

Step 4: Customer
4.1. Select a random number r'
4.2. Compute $b = r \cdot r'$
4.3. Send $\beta = b^e (u-x) \bmod \lambda$ to the broker

Step 5: Broker
5.1. Compute $\beta^{-1} \bmod \lambda$
$t = h(a)^d (\alpha(x^2+1)\beta^{-2})^{2d} \bmod \lambda$
5.2. Send (β^{-1}, t) to the customer

Step 6: Customer
6.1. Compute $c = (ux + 1)\cdot\beta^{-1}\cdot b^e = (ux+1)(u-x)^{-1} \bmod \lambda$
6.2. Compute $s = t \cdot r^2 \cdot (r')^4 \bmod \lambda$
The tuple (a, c, s) is the signature on message w_0. Any one can verify this signature by checking whether $s^e \equiv h(a)h(w_0)^2(c^2+1)^2 \bmod \lambda$.

Correctness: $s^e \equiv (t \cdot r^2 \cdot (r')^4)^e \bmod \lambda$
$\equiv (h(a)^d(\alpha(x^2+1)\beta^{-2})^{2d} r^2 (r')^4)^e \bmod \lambda$
$\equiv (h(a)^d(r^e h(w_0)(u^2+1)(x^2+1) r^{-2e} (r')^{-2e}(u-x)^{-2})^{2d} r^2 (r')^4)^e \bmod \lambda$
$\equiv (h(a)^d(r^e h(w_0)(u^2x^2+x^2+u^2+1) r^{-2e} (r')^{-2e}(u-x)^{-2})^{2d} r^2 (r')^4)^e \bmod \lambda$
$\equiv (h(a)^d(r^e h(w_0)((ux+1)^2+(u-x)^2) r^{-2e} r^{-2e}(u-x)^{-2})^{2d} r^2 (r')^4)^e \bmod \lambda$
$\equiv (h(a)^d(r^e h(w_0)(c^2+1) r^{-2e} (r')^{-2e})^{2d} r^2 (r')^4)^e \bmod \lambda$
$\equiv h(a)h(w_0)^2(c^2+1)^2 \bmod \lambda$

4 Analysis of the Improved Scheme

This section has two subsections, Security and Efficiency analysis, as explained below.
4.1 Security Analysis
The improved phase resists the following possible threats:

Unforgeability: An adversary cannot derive forged signatures. To successfully pass the verification equation $s^e \equiv h(a)h(w_0)^2(c^2+1)^2 \bmod \lambda$, the adversary has to compute s such that $s \equiv h(a)^d h(w_0)^{2d}(c^2+1)^{2d} \bmod \lambda$, given the values $h(a)$, $h(w_0)$ and c. However, it is computationally infeasible to acquire the value of d without the factorization of λ, which is an intractable problem. On the other hand, given s, $h(a)$ and $h(w_0)$, it is infeasible to compute c such that $c^2 \equiv (s^e \cdot h(a)^{-1} \cdot h(w_0)^{-2})^{1/2}-1 \bmod \lambda$ without factoring of λ. Given a and c, the adversary is unable to acquire s' such that $s' \equiv s \cdot h(w_0)^{-2d} h(w_0')^{2d} \bmod \lambda$ without knowing d. Without factoring λ, it is infeasible to get c' such that $c'^2 \equiv (s^e \cdot h(a)^{-1} \cdot h(w_0')^{-2})^{1/2}-1 \bmod \lambda$. It is also difficult to derive message w_0' with $w_0' \equiv w_0 \bmod \lambda$ such that $h(w_0) \equiv h(w_0') \bmod \lambda$, as $h(\cdot)$ is a cryptographically secure hash function. Therefore, the adversary cannot succeed to forge the signature.

10

Unlikability

For any given valid signature (a, c, s), no one except the requester is able to link the signature to its previous signing instance. This implies that the signer is unable to find the link between the signature and its corresponding signing process instance.
Randomization: The signer randomizes the blinded data using a random factor x before signing it in the signing phase. This randomization property keeps the blind signature scheme away from some chosen text attacks.

Double Spending Detection: Our scheme adopts the same transaction phase as in Section 2.3. U sends $\{R_{UM}$, (*a*, *c*, *s*), w_0, (w_j , *t*), s_k, *OI*, Expire$\}K_{UM}$ to M before taking service from M. The payment root R_{UM} is equal to $h(w_j \oplus (s_k \| K_{UM}))$. We note that the s_k, K_{UM} will be different in every purchase. As a consequence, B would be able to detect double spent paywords if U expends double the paywords.

Forgery Prevention: U obtains B's signature on w_0 before any transaction. The blind signature is based on RSA algorithm, which is widely used a secure signature algorithm. Besides, in order to process a correct redemption, M must have knowledge of the payment information. It is practically impossible for someone to forge U's paywords without knowing the secret key K_{UM} and K_{MB}.

Multiple Payment: In the transaction phase, U sends a request to B to get K_{UM} and creates the payment root $R_{UM} = h(w_j \oplus (s_k \| K_{UM}))$ where w_j is the first unused payword in the paywords sequence. Consequently, every time when U makes a purchase, the R_{UM} is not the same, which enables U to make payments with multiple merchants.

4.2 Efficiency

In a micro-payment system, the profit gained by a merchant is small in each transaction. It is unwise to verify the transaction using a complicated method that leads the average cost of the system exceeding the profit. In other words, large computation in micro-payment is not advisable. In order to measure efficiency of our improvement, we compare the improved blinding phase with the Hwang and Sung's scheme [8]. The computational complexity of the remaining phases remains same in both schemes. We use the following notation to measure the efficiency of the schemes.

t_h: computation time for hash operation
t_a: computation time for modular multiplication (or point addition in elliptic curve)
t_e: computation time for modular exponentiation (or scalar multiplication of a point on ellipt curve)
t_s: computation time for symmetric key encryption

As modular exponentiation is an expensive operation in comparisons with addition or hash operation, it is easy to see from Table-2 that our improvement is efficient than Hwang and Sung's scheme. Moreover, if one selects low public exponent e, say 3, then our scheme becomes more efficient. This makes public key operations faster while leaving private key operations unchanged. In that case, if one employs the low public exponent attack, s/he cannot succeed to this attempt because each signature is being randomized by some random factors. Therefore, the improved scheme reduces expensive exponential operation and achieves computational efficiency.

Table 2: Computational complexity in blinding phase

	Blinding Phase
Hwang and Sung scheme [8]	$4t_h + 7t_e + 3t_a + 1t_s$
Improved scheme	$2t_h + 6t_e$

Canopus Solution

As complex and far-reaching as the aforementioned problems are, the solution to the fundraising problem is quite simple. Canopus will offer the following features:

- Low volatility
- Safe Applications
- Faster completion
- Intuitive design
- Less expensive transactions with higher throughput
- Ability to move assets between blockchains

While the solution actually looks pretty elegant on paper, it was shackled by the available technology, right up to the Avalanche.
Canopus is a decentralized galactic network protocol that will offer token pools and auctions, allowing teams to raise funds for their projects in a permissionless and compliant environment using the Avalanche blockchain.
We will allow the creation and ownership of swap pools based on stable and flexible token prices. Our Pools offer projects several advantages over current models, for example:

- Stable price
- Predictable results
- More informed investors
- Reliable and compatible
- Fair distribution of tokens
- Positive community opinion
- In addition to token sale - OTC trading, auctions, whitelisted sales

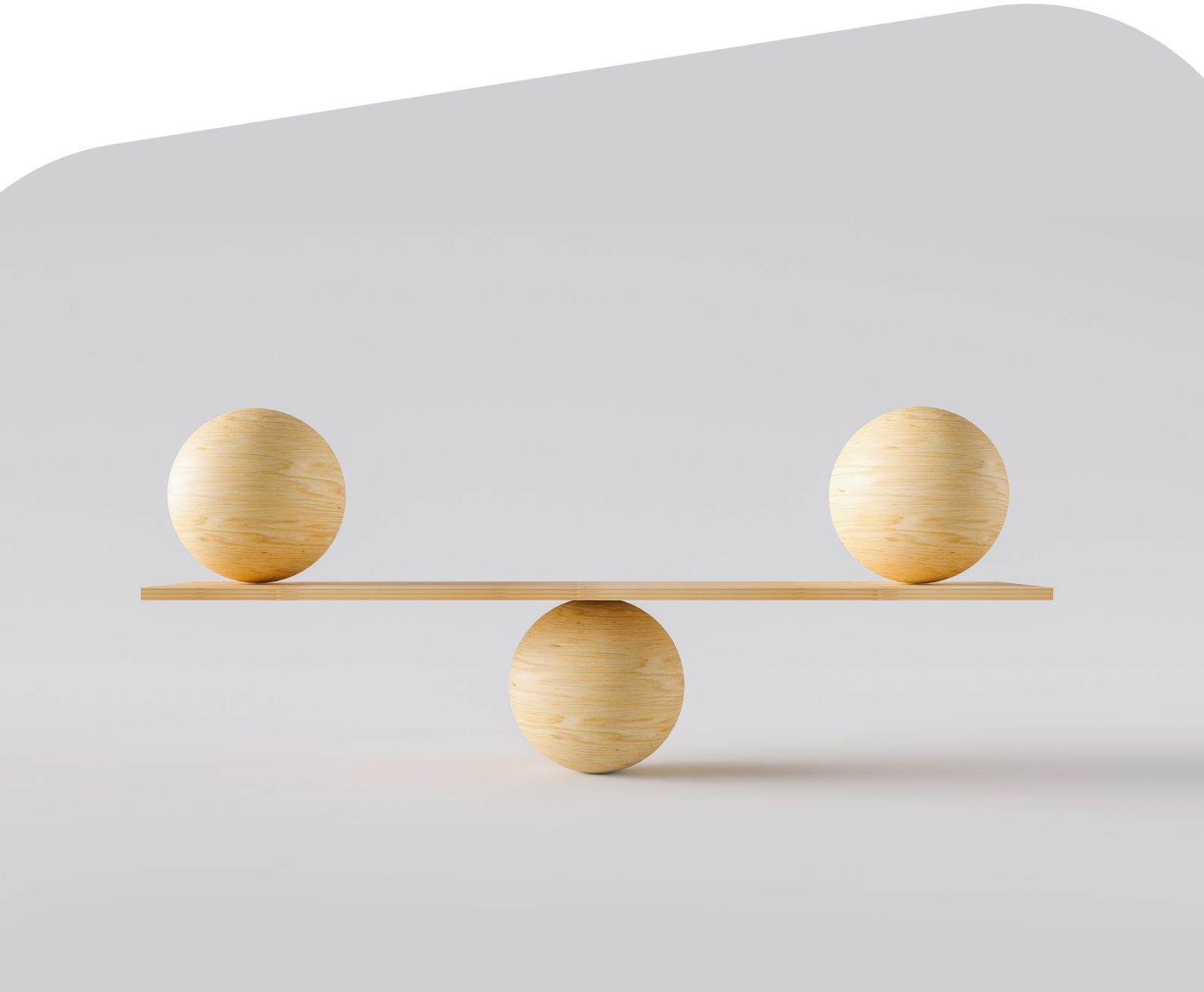
12

Conclusions

Ultimately, secure, fast and transparent fundraising that empowers the right teams will help move the space forward. Canopus is an investor-focused platform that supports rather than confuses members by offering an environment in which everyone wins. Our culture is built on rewarding innovation and talented teams that deliver real value through rewarding apps and forward-thinking development.

Token Specification

Name: Canopus Symbol: OPUS
Network: Avalanche C-Chain Spec: ARC20
Precision: 18
Total Supply: 1 Billion
Smart Contract Address: 0x76076880e1EBBcE597e6E15c47386cd34de4930F



Goverence

The Canopus (OPUS) governance model provides a transparent and fair listing process that targets innovative projects with strong perspectives and strong teams, giving projects the ability to access the exclusive Galactic cross-chain.

The fundamental core of the Canopus (OPUS)governance model is that users are given the opportunity to support the projects they want to see on Canopus with proposed projects that have been carefully evaluated and reviewed by leading finance, marketing and legal professionals. This process strictly allows only the strongest and most legitimate projects with the greatest potential to participate. With its growing popularity across multiple chains, Canopus (OPUS) provides a unique opportunity for projects in leading chains to reach a wider audience, raise capital and allow these projects to gain significantly greater impact across multiple chains previously unavailable with a combined result. overall positive growth.

We begin our work on Canopus' economic design with the governance discussion as it plays a critical role in future components. To enable the system to adapt to changing economic conditions, the Canopus platform allows basic system parameters to change dynamically based on user input. A viable process for finding globally acceptable system parameter values is critical for decentralized systems with no custodians. Avalanche can use the consensus mechanism to create a system that allows anyone to propose private transactions that are essentially system-wide surveys. Any participating node can make such offers.

The reward rate is an important parameter that affects any currency, whether digital or paper. Unfortunately, cryptocurrencies that fix this setting can face various problems, including deflation or hyperinflation. For this purpose, the amount of the fee will depend on the management, within predetermined limits. This will allow token holders to choose the speed at which OPUS reaches its limited supply.

Minting function

Minting on OPUS is designed to encourage nodes to act in a way that contributes positively to global results. This is achieved through special coin transactions. A node earns the right to build a mint by first setting a rate and then actively participating in the consensus building process. In particular, node rewards are directly related to uptime and response delays. Each node stores local information about the health and behavior of all other nodes with which it interacts. Whenever node v is selected by the U, the latter holds a local tuple (response bit, timestamp). The first entry is the only bit that indicates whether v is responding during timeout, and the second represents the timestamp of the response. In other words, chasing Avalanche is done through health and intervention control. This mechanism has important implications. In particular, there is no "get-rich-rich" added effect, as there is no "leader" that accumulates rewards.

14

Spam Management:

Although there is a fee for simple payments, their cost is almost zero. However, this can lead to spam on the network. In future releases, each transaction will have a local PoW to avoid congestion. PoW is inherently low in complexity and therefore a transaction can be performed immediately with very little overhead. However, if a particular key generates a large number of transactions in a short period of time, each subsequent action will present great difficulties in the PoW puzzle. This mechanism works in conjunction with wood burning.

References

Canopus smart contract code:
https://github.com/Canopus-Network-OPUS/OPUS/blob/main/ARC20

Github:
https://github.com/Canopus-Network-OPUS

Twitter:
https://twitter.com/Canopus_network

Telegram Channel/Group:
https://t.me/Canopus_Channel https://t.me/Canopus_Group

Instagram:
https://instagram.com/canopus_network

Discord:
https://discord.gg/8tck6BFa

Offical Email:
info@canopus.network

Offical Website:
https://canopus.network

Chain Explorer:
https://cchain.explorer.avax.network/address/0x76076880e1EBBcE597e6E15c47386cd34de4930F/transactions

Exchange Platform:
https://app.pangolin.exchange/#/swap?inputCurrency=0x76076880e1ebbce597e6e15c47386cd34de4930f

Staking:
https://app.pangolin.exchange/#/add/0x76076880e1ebbce597e6e15c47386cd34de4930f/AVAX

canopus ROADMAP

Q1

- SMART CONTRACT AVALANCHE C-CHAIN
- LAUNCH/IDO

Q2

- IMPROVING SECURITY AND STABILITY
- SWAP CROSS-CHAIN
- STAKING
- IMPROVING THE TRANSACTION TIME AND FEE
- MARKETING CAMPAIGN
- TOKEN DISTRIBUTION
- INTEGRATE WITH EXCHANGE PLATFORMS

Q3

- WEB WALLET
- STABLE POOL
- LISTING
- FUTURES CONTRACTS
- SYNTHETIC ASSETS
- SECURITIES

Q4

- COMMUNITY APP
- P2P INTEGRATION
- INTEGRATE WITH E-COMMERCE PLATFORMS
- BUILDING EXCHANGE PLATFORMS / WEB & APP