Volume: 16 Issue: 02 Pages: 6318– 6327 (2024) ISSN: 0975-0290

A Study of New Micropayment Scheme for Blockchain

Gulab Das
Department of Mathematics, Govt.N. PG College of Science, Raipur, Chhattisgarh, India.
Email: gulabdas03@gmail.com
B. P. Tripathi
Department of Mathematics, Govt.N. PG College of Science, Raipur, Chhattisgarh, India.
Email: bhanutripathi@gmail.com

ABSTRACT

Cryptocurrencies based on blockchain infrastructures have shown their advantages such as double-spending resistance and decentralization. Each transaction of cryptocurrency requires a certain amount of computation and attracts transaction fees. Often, in practice, many transactions are small; therefore, they add computation and transmission overheads to the system. In this paper, we introduce a cost-saving approach, which significantly reduces transaction time and storage for small amount of payment, i.e. micropayment. Micropayment means the value of transaction is small, i.e., payment worths a few pennies. To achieve instant micropayments, Hearn and Spilman introduced a notion of payment channel. In this paper, we formally discuss the robustness requirements of a scheme that is suitable for micropayments, consider the explicit value of penalty and user privacy leakage. More precisely, we propose a micropayments scheme for decentralized blockchain-based payment system based on the notion of payment channel, which enables a payee to receive funds at several unsynchronized points of sale and penalize the double-spenders, with instant confirmation. In our approach, with the notion of 'transaction commitment', the computation of each transaction is much more efficient. Therefore, our approach has advantages in comparison of other cryptocurrency systems such as the bitcoin system. Our approach can be applied to other existing cryptocurrency systems.

Keywords – **Bitcoin, Block Chain, Double-spending, Micropayments, User privacy.**

Date of Submission: May 19, 2024 Date of Acceptance: June 16, 2024

1. Introduction

Traditionally, payments are paper-based and usually performed among a payer, a merchant and a bank. The transactions are verified through the paper-based signature. Nowadays, the electronic payments have been widely adopted. Electronic payments attract the computational cost, regardless the transaction is large or small. In the practice, many transactions are rather small (or micropayment). The transaction cost could become even higher than the value of the micropayment. In order to solve this problem, efficient micropayment systems have been proposed.

Electronic payment approaches have exhibited many advantages incomparison with paper-based payment approaches. Electronic payment approaches are usually based on cryptographic tools, which ensure the security and privacy of payment. Cryptographic tools are the foundations to ensure that transactions are non-reservable, which prevents users from fraud. However, electronic payments rely on a trusted server who handles transactions. This is often regarded as undesirable, as a single failure of trust might compromise the system.

Early approaches of electronic payment stem from Digital Cash[1], which can be spent like paper-based cash with the feature of anonymity of users. The problem with the user anonymity is double spending, since the double spender cannot be found due to its anonymity. With the aid of blind signature [2], the problem has been solved. However, it relies on the trust of the bank which manages the electronic payment system. Due to a number of reasons, the idea of digital cash has not been well accepted by the community and we have not seen a truely successful digital cash system.

Satoshi Nakamoto in 2008 [3] proposed a completely new idea of peer-to-peer electronic payment system (bitcoin). This idea is based on the peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The notion of blockchain was therefore introduced to the electronic payment. The blockchain is a peer-to-peer network which can prevent double-spending in the application of electronic cash. An electronic coin is represented by a chain of digital signatures in which each owner transfers its coin to the next one by signing the hash of the previous transaction (transaction history) and the public key of the next owner (or the address). Double-spending resistance is due to the fact that 'the network timestamps transactions by hashing them into an ongoing chain of hash-based proof-ofwork (PoW)'. The blockchain therefore consists of a distributed ordered list of transactions which cannot be changed without redoing the proof-of-work. Bitcoin is the world's first completely decentralized cryptocurrency. The invention of bitcoin is revolutionary, because for the first time, the double spending problem can be solved without the need for a trusted third party. The key to solve the double spending problem is to allow the ledger to be distributed to all the users of the system via a peer-to-peer

Volume: 16 Issue: 02 Pages: 6318– 6327 (2024) ISSN: 0975-0290

network. All transactions are recorded in the public distributed ledger called blockchain. The information added to the blockchain becomes harder to revoke as many blocks are added to the public ledger and any new block creation is costly, due to PoW. Any new transaction can be checked against the blockchain to ensure it has not been previously spent. The global peer-to-peer network is composed of thousands of users who ensure the functionality of blockchain without need of any trusted third party.

The operation of bitcoin transactions relies on users called miners, who are responsible for verifying and reconciling the ledger and are rewarded for their work. To receive a reward for a transaction, miners need to solve a mathematical puzzle. The process of solving such a puzzle is called PoW. A large number of small transactions will significantly reduce the transaction speed due to the required computation and verification of a transaction, regardless of small or large.

There are some concerns on bitcoin and blockchain including scalability and speed of transactions, storage, processing, latency and bandwidth. The limitations of bitcoin have impacted on users due to the delay in processing transactions and growth of transaction fees. These fees can elevate as the competition for space in the blockchain increases [4]. In this case, the micropayment could significantly consume the computational resources.

Micropayments are referred to the payment schemes which enable the payments of small amounts. For example, they can be applied for payments of each visit to websites or other online services [5]. A micropayment channel can also provide transactions among two parties off-blockchain, without contacting the blockchain, with smart contracts [6], users might ignore using the smart contracts by committing to the total balance in a transaction [7]. However, it is prominent to provide efficient micropayment to reduce the cost of operations in terms of the value of payments, and therefore the delivery of payments is of importance, unless a large scale or persistent fraud is detected[8].

Despite the efforts of making efficient micropayment schemes with the blockchain technology, there are still some issues regarding the cost in micropayment. For example, in the lightning network [9], the micropayment transactions are required to be signed and verified by users. In this paper, we introduce a new approach of micropayment based on blockchain and cryptographic tools. Our scheme solves the problem of transaction speed due to the computation and verification of a transaction. The feature of our scheme lies in the fact that the generation of verification of microcoins do not require any digital signature. Intuitionally, in our scheme the payer commits a total amount of payment to the payee, say an amount of bitcoin X, and then the payer can generate 'microcoins' from X and pay to the payee with the microcoins for each micropayment transaction. An example of this scenario could be: the shop which sales videos. The payer can subscribe to the shop for one week. For each day, the payer will pay some amount of microcoins until the end of the week. Our scheme has exhibited the following features:

•Since all microcoins are linked to the commitment, the verification of a microcoin is very efficient based on hashing.

• For each microcoin, only need to store a hash value.

• No any signing is required for each transaction.

• Save the communication cost during a micropayment. The process of generation of transactions of our microcoins is not trivial due to the property of one-wayness and collision resistance of traditional hash functions. In our scheme, we adopt the tool of chameleon hash function and incorporate it with our micro hash chain. We define the adversarial model for our scheme and prove its security according to the model. In regard to the implementation of our scheme, our scheme can be based on bitcoin. Therefore, the only change is allowing micropayment. We can divide the micropayment scheme into two phases:

• In the commitment phase, the commitment is verified by miners with the same procedure of verification of a transaction in bitcoin. The only change is the cryptographic algorithm to capture the commitment part.

• In the payment phase, the payer follows the micropayment protocol, which allows microcoins to be spent and verified within a local coin-based chain. There are two main techniques to handle transactions with small value. Payment channel[10]is emerging in bitcoin community [9, 11]that needs two transactions being confirmed in blockchain network: creating channel transaction and closing channel transaction. P robabilistic payment[12]lets payee receive a macro-value with a given probability and a micro-value for each transaction in expectation.

Decentralized Micropayments. In decentralized system, all participants achieve an agreement together via consensus mechanism, i.e., proofs-of-work. Realizing micropayments in decentralized system brings us new challenge to balance efficiency and security.

Double Spending. Micropayments scheme, which requires payee responding to payer in short time and just doing local confirmation, is easy suffering from double-spending attack that payer reuses a valid voucher cert to different unsynchronized payees repeatedly before being detected.

User Privacy. User privacy is not only concealment of identity, such as the pseudonym in bitcoin system. In this work, we also consider protecting user transaction message among the unsynchronized points of sale. We here ask the following question:

Is that possible to strengthen micropayments scheme for decentralized blockchainbased payment system so that it can be secure even adversary reuses a voucher repeatedly before being detected and enhance user privacy among the unsynchronized points of sale?

1.1 Our Contributions

We give an affirmative answer to the above question. Most existing micropayments schemes[5,12,13,14]focus on general setting, where payee is a single entity. In real word, it is usual that a merchant consists of several geographically distributed and unsynchronized points of sale. We mainly focus on the security of micropayments scheme in this complex setting.

Volume: 16 Issue: 02 Pages: 6318– 6327 (2024) ISSN: 0975-0290

General Setting. The first step, we assume that payee B is a single entity and accepts small payments as shown in micropayment 1.
Complex Setting. Based on step one, We go further to explore a complex setting[15]as shown in micropayment 2 where B consists of several geographically distributed and unsynchronized points of sale. We then propose our construction micropayment 3 that solves the security problems of micropayment 2.

Robustness Requirements for Achieving Micropayments:
1. Basic requirements in general setting:
(i)**Instant Confirmation**. Micropayment requires quick response, i.e., payer will receive service as soon as he sends valid messages (voucher).
(ii)**Small Transaction Fees**. .Payer is unwilling to use a payments system to handle small transactions, which costs high transaction fess, since the fees maybe higher than the value of transaction.
2. Additional requirements in complex setting:
(i)**Preventing Double-Spending**. Security in the presence of reusing a voucher (cert), i.e., payer spends a cert to different points of sale with the risk of being detected and losing coins.
(ii)**Protecting User Privacy**. Security in the presence of using the voucher provided by the last transaction in the current transaction, i.e., payer spends a voucher cert signed by last payee to current payee without disclosing the identity of last payee.
Expiry Time. To solve the above questions, we propose a notion of expiry time, which means that each voucher is valid during a given time.
Upper Bound of Penalty. We give a proper value of penalty, which means that it makes the malicious payer at a disadvantage for his dishonesty and is reasonable for honest payer. What's more, we get the upper bound of penalty as $p \leq \hat{T}/\tilde{T} * u_2$ (more details are in Sect. 3.3).
For user privacy, we utilize ring signature during the process of paying through channel to break linkage between singer and signature.

1.2 Outline of the Paper

The remaining sections of this paper are arranged as follows. In Section 2, we describe some related work about micropayment. In Section 3, we provide some preliminary tools which are required in our scheme. In Section 4,we discussed about micropayment system .In Section 5, we present our system model to illustrate how our system should be constructed. In Section 6, we define our micropayment scheme by defining our algorithms and adversarial model. In Section 7, we present the detailed algorithms and construction of micropayment of our scheme. In Section 8, we present the security analysis of our scheme . In Section 9, we conclude our work.

2. Related Work

Micropayment research can be stemmed to decades ago. The original micropayment research was conducted with the intention to reduce the transaction fees of credit card based payment systems. There are a number of micropayment schemes in the literature, including[5,8,16,17]. All these micropayment schemes are in general based on hashing chains, where each block in a hashing chain is regarded as a coin with some monetary unit. These coins can be spent while needed. The verification of these coins relies on the root of the hashing chain which has usually been authenticated.

Manasse et al.[18] created lightweight protocols for electronic commerce that support purchases under a cent with the use of brokers to manage accounts, and script as a valid digital cash. However, they avoided using any encryption Then, Rivest and Shamir [8] created two micropayment schemes, namely 'PayWord' and 'MicroMint', to minimize the number of public-key operations required per transaction using the hash function. The 'Payword' is a creditbased scheme, which the user needs to authenticate the complete chain to the vendor with a single public-key signature, and then reveal each PayWord in the chain to the vendor to make the micropayments. However, 'MicroMint' has been designed to eliminate the necessity of public-key in order to speed up the operations. However, MicroMint are presented by hash function collisions which is based on birthday paradox. Following this, Shamir[17] proposed a micropayment scheme based on the use of probabilistic payments with 'electronic lottery tickets'. The proposed notion is much more efficient where the bank or broker does not have to follow the conventional payment schemes to process each payment, and only the 'winners' are redeemed.

Lipton and Ostrovsky [19] proposed a protocol for 'Duplex Micropayment Channels' that ensures end-to-end security and forms a network of payment service providers (PSP). The authors tried to reduce the reliance on the blockchain, and only consider it if there is a need to publish long lived pointto-point channels between parties. The blockchain is therefore involved during the setup and the closure of the channel, and updates are not committed to the blockchain.
Pass and Shelat[12] mentioned that proposed micropayments rely on a trusted third party for coordinating transactions. Hereby, they proposed a new lotterybased micropayment scheme based on blockchain, and implemented in a web application. In a following work by McCorry et al. [4], fair exchange payment channels for off-chain transactions with the use of Hashed Time-Locked Contracts (HTLCs) have been proposed. They also compared Duplex Micropayment Channels and Lightning Channels, in terms of computation, storage and network access.

In[20], solutions to the problem of anonymity in bitcoin were proposed for transactions on bitcoin's blockchain and off the blockchain (micropayment channel networks). They used an honest-but-curious intermediary to issue anonymous vouchers and used bitcoin as a platform to confirm transactions. The blind signature has been used to achieve unlinkability. Burhcert et al. [7] considered the scalability issue of bitcoin networks. A trust-less off-

Volume: 16 Issue: 02 Pages: 6318– 6327 (2024) ISSN: 0975-0290

blockchain channel has been proposed to enable the connection between the blockchain and the payment channels. They argued that the cost of transactions in the blockchain can be reduced compared to regular micropayment channels. A recent work about efficient payment for the cloud computing was proposed [21]. In this paper, we introduce an entirely new way to blockchain-based micropayment, with the aim of reducing transaction costs and improving transaction speed. Our method offers a new approach for micropayment.

Many off-line micropayments schemes are proposed[14,17,22] with a trusted third party to sign a voucher for payer and punish cheaters. Bitcoin system is a peer-to-peer fully decentralized payment system introduced in[3]. Unlike traditional e-cash system [23], where there is a central bank to handle transactions and detect cheaters. Decentralized system utilizes distributed public ledger blockchain to record all transactions.

Probabilistic payment was proposed in[5,17]that allows payer to execute series of small transactions. Rivest[17] and Micali[5] proposed lottery-based payment to overcome the relative high fees of small transactions.[5,12]are implementations of this idea. [13] presents a decentralized micropayment scheme by following the way of probability payment. Creating payment channel was introduced in[10].[24]discusses two major questions about why we need micropayments. Further studies as[9,25]. Constructing anonymity set[26]enhances privacy in some certain situations. [20] uses TumbleBit, a new unidirectional unlinkable payment hub, to allow payer to execute payment via an untrusted intermediary. These schemes are secure if the size of set is big enough and majority of participants are alive.[27]proposed a micropayment scheme in complex setting, but there are two problems obviously in this scheme: double-spending and user privacy leakage. More details are in Sect. 3.2.

3. Preleminaries

In this section, we revisit some useful tools for our scheme. We give the main techniques behind our construction and the definitions of security properties are presented in game-based fashion.

3.1 Techniques

Definition 1(Ring Signature).A ring signature scheme is a triple of p.p.t. algorithms algorithms RS = (Gen, Sign, V rfy) [28]. Formally:

–Gen(1^{λ}).Takes as input the security parameter λ, outputs a public key pk and a secret key sk.

–$Sign_{sk}$(R, M). Outputs a signature σ on message M with respect to ring R = (pk_1, ..., pk_n).

–VrfyR(M, σ).Takes as input a ring R, a message m, and a signature σ for M to return a single bit b = 1/0.

Definition 2(Accountable Assertion). We recall the definition in[28]that consists of four algorithms Q = (Gen, Assert, Verify, Extract) :

–(pk, sk, auxsk) ← Gen(1^{λ}):Outputs a key pair consisting of a public key pk and a secret key sk, and auxiliary secret information auxsk.

τ/⊥ ← Assert(sk, auxsk, ct, st) : Takes as input a secret key sk, auxiliary secret information auxsk, a context ct, and a statement st and returns either an assertion τ or ⊥ to indicate failure.

–b ← Verify(pk, ct, st, τ) : Outputs 1 if τ is a valid assertion of a statement st in the context ct under the public key pk.

sk/⊥ ← Extract(pk, ct, st_0, st_1, τ_0, τ_1):Takes as input a public key pk, a context ct, two statements st_0, st_1, two assertions τ_0, τ_1 and returns either the secret key sk or ⊥ to indicate failure.

3.2 Hash Function

Cryptographic hash is a family of functions H : $\{0, 1\}^{*} \rightarrow \{0, 1\}^{l}$ compressing bit-strings of arbitrary length to bitstrings of a fixed length l. A secure cryptographic hash function should have the following properties: (1) one-wayness: the function should be easy to compute; (2) collision resistance: it is computationally infeasible, given one of these functions H, to find a pair of distinct strings x,´x satisfying H(x) = H(´x).

3.3 Chameleon hash functions

Chameleon hash functions are a type of collision resistant hash functions, which is associated with a pair of public and private keys, where the private key is called the trapdoor. This type of hash function has the following properties: (1) anyone can compute the associated hash function with the corresponding public key; (2) it is collision resistant, provided the trapdoor is not known; (3) the owner of the trapdoor can easily find collisions of many given input values.

Intuitionally, a chameleon hash function is associated with a user U who has a public key pkU and holds a corresponding secret key denoted by skU .The public key pkU defines a chameleon hash function, denoted by CHASHU (., .) which can be computed efficiently with given the value of pkU . On input a value m and a random string r, the chameleon hash function generates a hash value CHASHU (m, r) which meets the following properties [29]:

•Collision resistance: There is no efficient algorithm that on input the public key hkU can find pairs (m_1, r_1) and (m_2, r_2) where $m_1 \neq m_2$ such that CHASHU (m_1, r_1) = CHASHU (m_2, r_2).

• Trapdoor collisions: There is an efficient algorithm that on input the secret key skU , any pair (m_1, r_1) and any additional value m_2, finds a value r_2 such that CHASHU (m_1, r_1) = CHASHU (m_2, r_2).

• Uniformity: All values m induce the same probability distribution on CHASHU (m, r) for r chosen uniformly at random. This condition can be relaxed to require that the above distributions are not necessarily identical for all values be computationally indistinguishable [29].

3.4 Digital signature

Digital signature is a useful tool to achieve authenticity of a message, based on public-key cryptography. Given a pair of public–private keys (pk, sk) of a user, say Alice, Alice signs a message m with her private key sk. The receiver of the signature, say Bob, can verify the signature with Alice's public key pk. The security of digital signatures has been well defined and can be mathematically proven. We will give the security definition of digital signatures later.

Volume: 16 Issue: 02 Pages: 6318– 6327 (2024) ISSN: 0975-0290

3.5 Security Properties

According to our goals, the micropayments scheme should satisfy three security properties: unforgeability, unlinkability and double-spending detection. We show these security properties in the following three experiments as $Exp\Pi_m^{uf}, A^{\lambda}$, $Exp\Pi_m^{ui}, A^{\lambda}$ and $Exp\Pi_m^{ds}, A^{\lambda}$.

Definition 3(Unforgeability, Unlinkability, double-spending detection). Given a micropayments scheme Π^m in blockchain-based system, a p.p.t. adversary A, security parameter λ and consider the followings:

Experiment $\mathbf{Exp}\Pi_m^{uf}, \mathbf{A}^{\lambda}$

$\{(pki, ski)\}_{i=1}^{n} \leftarrow RS.Gen(1^{\lambda})$; Q = Φ; R = $\{(pki)\}_{i=1}^{n}$ cert∗ ← $A^{Ocert(i,R,state)}$ (R); i∈[n] is index of each sale Q = Q ∪(., R, state), cert∗ = (state∗ , σ∗); (., R, state∗) does not belongs to Q if VefyR(state∗ , σ∗) = 1, then return 1, else return 0.

Experiment $\mathbf{Exp}\Pi_m^{ul}, \mathbf{A}^{\lambda}$

$\{(pki, ski)\}_{i=1}^{n} \leftarrow RS.Gen(1^{\lambda})$; Q = Φ; R = $\{(pki)\}_{i=1}^{n}$ (cert∗ , R, i) ← $^{AOlink(cert,R)}$ (R); i∈[n] is index of each sale Q = Q ∪ (cert, R); (cert∗ , R) does not belongs to Q if Bi is the signer of cert∗ , then return 1, else return 0

Experiment $\mathbf{Exp}\Pi_m^{ds}, \mathbf{A}^{\lambda}$

$\{(pki, ski)\}_{i=1}^{n} \leftarrow RS.RKGen(1^{\lambda})$;

(pkA, skA, auxskA) ← Π.Gen(1λ) Q = Φ; R = $\{(pki)\}_{i=1}^{n}$; (service, cert′) ← $^{AOspend(tx,\tau,cert)}$ (R) Q = Q∪ (tx, τ, cert); skA ← Extract(pkA, Q) if {(tx, τ, cert),(tx′ , τ ′ , cert)} ∈ Q∧(pkA, skA) does not belongs to Π.Gen(1^{λ}) then return 1, else return 0

We define the advantage of A in the above experiments as:

$\boldsymbol{Adv}\Pi_m^{ul}, \mathbf{A}^{\lambda}$ = P r[$\mathbf{Exp}\Pi_m^{ul}, \mathbf{A\lambda}$= 1] – 1/ n

$\boldsymbol{Adv}\Pi_m^{uf}, \mathbf{A}^{\lambda}$ = P r[$\mathbf{Exp}\Pi_m^{uf}, \mathbf{A\lambda}$= 1]

$\boldsymbol{Adv}\Pi_m^{ul}, \mathbf{A}^{\lambda}$ = P r[$\mathbf{Exp}\Pi_m^{ds}, \mathbf{A\lambda}$= 1]

4 Micropayments System

In this section, we propose a scheme about achieving micropayments in decentralized blockchain-based system in three steps.

4.1 Micropay 1

Before showing the description of micropay 1, we assume that A has a bitcoin address pk_1 with value υ and unforgeable digital signature scheme with algorithms (Gen, Sign, Vefy). We show this scheme in which A micropays to a single B as follows:

– Stage 1 creating a payment channel

• Set-up

∗ A generates new key-pairs (pk_{esc}, sk_{esc}) snd (pk_2, sk_2) for escrow transaction and revoking deposit after expiry time t respectively

. ∗ B generates a new key-pair (pk_3, sk_3) .

• Escrow transaction

∗ A transfers value d(d ≤ υ) from address (pk_1) to address (pk_{esc}) by transaction tx_{esc} = (y, π_{esc}, d, t) to create a payment channel with amount d and sets the release condition π as $\pi_{esc}(x) = 1$ if one of the following two conditions is true:

(1)x = (t[x_1], σ_{skesc} , σ_{sk3} , t) and V rfypk_{esc} (t[x_1], σ_{skesc}) = 1, $Vrfy_{pk3}$ (t[x_1], σ_{sk3}) = 1, current time T < t, where transaction tx_1 controlled by B.

(2) x = (t[x_2], σ_{sk2} , t) and $Vrfy_{pk2}$ (t[x_2], σ_{sk2}) = 1, current time T > t, where transaction tx_2 controlled by A.

∗B signs a voucher cert and sends it to A after transaction tx_{esc} is confirmed in bitcoin network, where cert = (state, σ) , σ = $Sign_{skB}$ (state) and state = (pk_{esc}, d, b = 0).

∗ A verifies cert with public key pk_B.

– Stage 2: paying through the channel

• A agrees to pay b_1 to B. A sends transaction tx = (y_{esc}, π, $b+b_1$, σ_{skesc}) and cert to B, where y_{esc} is the index of transaction tx_{esc}.

• B receives (tx^* , $cert^*$), parses $state^*$ = (pk_{esc}, d^* , b^*) and verifies the following conditions.

(1) (tx^* , $cert^*$) are valid,$cert^*$ has not been used before and $b^* + b_1 \leq d$

(2) pk_{esc} does not belongs to BL(A is not in blacklist), T < t and T is the current time

• B updates state as state = (pk_{esc}, d, b = $b^* + b_1$), signs state as σ, records (tx^* , $cert^*$) and sends (cert, service) to A.

– Stage 3: closing the channel

• B closes payment channel at one of the three conditions: (1)B detects that A reuses a cert and adds (pk_{esc}) to BL (2)b = d or d − b is too small to pay for a transaction (3)time t is reached

• A closes payment channel at the conditions:

∗ T > t (T is the current time) and B does not close channel

Security Analysis.

In micropay 1, A succeeds to micropay to B with one security problem that B can get all knowledge of A's purchase messages that breaks A's privacy.

4.2 Micropay 2

Now we show a scheme in which B is a distributed entity by recalling the construction in [30]. We give a simple description in which A micropays to a distributed B as follows:

Assumptions: B and its points of sale Bi have corresponding key pairs (pk_B, sk_B), $\{(\{(pkBi, skBi)\})\}_{i=1}^{n}$ respectively. B collects transactions recorded by each B_i at time T'

. **– Stage 1 creating a payment channel**

• A sets up bitcoin key pair (pk_B, sk_B) and accountable assertions keys (apk = pk_A, ask = sk_A, auxsk) for non-equivocation contracts.

• A creates payment channel with amount d + p and expiry time t(t > T′).

• B provides a signed voucher cert = (state, σ), wherestate = (t, d, k = 0, b =0, B), σ = $Sign_{sk}$B(state), after escrow transaction is confirmed in network.

– Stage 2: paying through the channel

• A agrees to pay bi to B_i , then B_i selects a fresh nonce r and sends it to A.

• A computes τ ← Assert(ask, auxsk, k, r) and sends (tx, τ, cert) to B_i.

• Bi receives (tx^* , τ^* , $cert^*$), parses $state^*$ = (t^* , d^*, k^* , b^* , B_j) and verifies:

∗ Vrfy(pk_{Bj} , $state^*$, σ^*) = 1, Verify(apk, k^* , r, τ^*) = 1

∗ tx^* is a valid transaction with amount $b^* + b_i$ and $b^* + b_i \leq d^*$

∗ A does not belongs to BL (A is not in blacklist) and T < t^* (T is current time)

Volume: 16 Issue: 02 Pages: 6318– 6327 (2024) ISSN: 0975-0290

• B_i updates $k = k^* + 1$, $b = b^* + b_i$, signs $\sigma = Sign_{sk}Bi(state)$, where state = ($t^*$, d^* , k, b, B_i) , tx^* ,τ^* and sends (service, cert) to A.

– Stage 3: closing the channel

• B collects all transactions recorded by each B_i at time T' and close the channel at one of the three conditions: ∗Expiry time t is reached and A is honest · B signs and broadcasts the last tx that is sent by A to get funds

∗ B detects A' dishonesty by τ · B extracts sk_A from two different assertions τ_1, τ_2 about k · B signs a transaction with d + p from payment channel with sk_B, sk_A ∗ b = d or d − b is too small to pay a transaction · B signs and broadcasts the last tx that is sent by A to get funds

• A signs a transaction with d+p from payment channel to closes the channel:

∗ T > t (T is the current time) and B does not close the channel

Security Analysis. In micropay 2, A succeeds to micropay to a distributed B, but with the following security problems:

(1)During **Stage 1**, it does not specify the size of p, so that A can spend more than d + p easily. For example, A reuses a cert signed by B many times and spends $\{\{bi\}\{(bi)\}_{i=1}^{n}|bi$ < d to $\{(Bi)\}_{i=1}^{n}$ in time T' . Consequently $\sum_{i=1}^{n} bi$ > d + p, which makes penalty useless.

(2)During **Stage 1**, A sends cert signed by B_j to B_i .So B_i verifies cert by doing $Vrfy(pk_{Bj}$, $cert^*$) = 1 and B_i gets knowledge that A has bought service from B_j , which breaks A's privacy.

4.3 Micropay 3

To overcome the problems in micropay 2, we present micropay 3. In this scheme, we employ expiry time to control the number of a cert being reused and use ring signature scheme to hide A's former purchase messages to current payee.

Notations. d, p is denoted the amount of deposit and penalty respectively. T is time that escrow transaction is locked and T' is the expiry time of voucher cert. Price of service provided by B_i is υ_i and we let $u_1 = \min\{\upsilon_i , ..., \upsilon_i\}$ and $u_2 = \max\{\upsilon_i , ..., \upsilon_i\}$.The average time of each transaction is denoted by $\tilde{T}$, $\bar{T}$ is time slot that B collects all transactions recorded by each point B_i and $\hat{T}$ is the working time of each point B_i within time $\bar{T}$. Let $T = l\bar{T} + T_{(conf)}(l = 1, 2, ...)$ to ensure that B can close the payment channel before A revokes escrow transaction and $T_{(conf)}$ is a safety margin to guarantee transactions broadcasted by B being confirmed on blockchain.

5 System Model

The micropayment system consists of a group of users who are involved in transactions and a blockchain-based cryptocurrency system for micropayment transactions. Users include the owners of digital coins or microcoins generated from a digital coin, receivers of microcoins, and miners who validate all transactions.

We will take the bitcoin system as an example to describe our system. Therefore, the setup of the system is similar to the bitcoin system. Each user holds a pair of private and public keys. Only additional requirement of the setup is that we require the key pair can be used for generating chameleon hash functions, apart from signing. This can be achieved easily under a discretelog-based crypto algorithm. The owner (on the left side) of the bitcoin generates microcoins from the bitcoin she owns. These microcoins are designated for another user (the receiver on the right side) only, who receives the microcoins from multiple transactions made by the original owner of microcoins. These microcoins are linked to the root which is generated directly from the bitcoin with the private key of the owner. The microcoins are linked to the root (we also call it 'commitment' of micropayment). Once the root is generated, miners in the system will validate it by PoW and record it in their ledgers if the validation returns true. The PoW defined in our scheme is still the same as that of Bitcoin. Only deference is about the verification of commitment, rather than verification of a signature only.

A transaction of microcoin starts from the microcoins next to the root. In case of a transaction of multiple microcoins, it must be ensured that the value of the prior transactions be deducted. The merit of this system lies in the feature that the microcoin transactions are very efficient in terms of computation.

The update of ledgers for micropayment transactions starts from the owner and the receiver. They directly update their ledgers and broadcast the updated information to the network. Other users will update their ledgers after a simple verification of validity of the micropayment. As microcoins are linked to the root which has been verified by miners, the verification of microcoins is simply based on hashing to the root, which can be computed very efficiently. The reader might wonder how such a chain of microcoins is generated. The trick here is chameleon hashing. We utilize the collision feature to connect the microcoins chain to the root, which also ensures the security of the chain. To do it, the owner needs the sole trapdoor key which is used to find a collision. This trapdoor key is also the private signing key; therefore, there is only a small change in the system setup phase.

The major change is to add the functionality of computation and verification of chameleon functions. Miners are required to validate the correctness of the collision of the chameleon functions along with the digital signature for the transaction.

6 Definitions

In this section, we present the definitions of our micropayment algorithms and the adversarial model for the later security analysis

6.1 Definitions of micropayment algorithms

Our micropayment system consists of the following algorithms

KeyGen(l^{λ}): Taking as an input a security parameter l, the algorithm returns a pair of public and private keys (pk, sk) . Sign(sk ,m): Taking as an input the private key sk and a message m, the algorithm returns a signature σ on m.

Volume: 16 Issue: 02 Pages: 6318– 6327 (2024) ISSN: 0975-0290

Verify(σ,pk,m) : Taking as an input a signature σ, the corresponding public key pk and the corresponding message m, the algorithm returns true or false.
CHashGen(pkr,,m): Taking as an input the public key pk, a random value r and a message m, the algorithm returns a value of chameleon hash function h.
CollComp(sk,pk,r,m,m′): Taking as an input a private key sk, the corre sponding public key pk, a random value r, messages m, m', where pk, r, m define a chameleon function value h, the algorithm returns a value r' and a new value of chameleon hash function h' such that h = h', where pk, r', m' , defines the new hash value h'.
MicroCoinGen(b,sk,pk,n) : Taking as an input a bitcoin value b, a pri vate key sk, the corresponding public key pk, and a committed number n of microcoins, the algorithm returns a chain of n microcoins ci and auxiliary in formation Aux, or i = 1,2,...,n. denoted by c. Aux contains the parameters which have been used to generate the commitment and microcoins.
Transact(cj,Aux) : Taking as an input a microcoin cj and the auxiliary in formation Aux, the algorithm returns true or false.

The adversarial model defines two types of adversaries—Types I and II.

• Type I: This type of adversary is referred to the owner of the crypto coin, who intends to spend more than what it has during a transaction

. •Type II: This type of adversary is referred to the receiver of a transaction, who intends to claim more than what it has received.

DEFINITION 6.1 (Type I Adversary). Equipped with a pair of private and public keys wrt a crypto coin b and suppose that the blockchain system is secure (i.e. assuming the integrity of ledgers), given a complete set of n micro coins (c_0,c_1,..., c_n) generated from b, it is computationally infeasible for the adversary A to generate any valid microcoin $c_{n'}$ such that n' ¿ n.

It is modeled with the following game between the adversary A and the sim ulator B

Setup: The simulator B calls the algorithm KeyGen(l^{λ}) to generate a pair of public and private keys (pk, sk) which are given to the adversary A Query: Given a bitcoin b, the adversary A can generate as many sets of mi crocoins as possible, where these sets of microcoins are denoted by Z.
Challenge: The challenger B selects a set of microcoins from Z and sends it to A . We denote this set of microcoins by (c^*_0,c^*_1,...,c^*_n).A wins the game, if A can return a valid microcoin cn′ such that n'¿n.
DEFINITION 6.2 (Type II Adversary). Given a microcoin ci, it is computationally infeasible for the adversary to generate a new microcoin cj such that it can pass the verification process.

It is modeled with the following game between the adversary A and the sim ulator B

Setup: The simulator B calls the algorithm KeyGen(lλ) to generate a pair of public and private keys (pk, sk) and gives the public key pk to the adversary A.
Query: The adversary can make at most n- 1 queries to B for at most n 1 microcoins from a given set of microcoins generated from the bitcoin b of (c_0,c_1,..., c_n) .These queries must be made in order; hence there is at least one valid microcoin, which has not been queried.
Challenge: Suppose the last query made by A is c_l. The challenger B selects a number which is greater then l and sends it to A . A wins the game, if A can return a valid microcoin cl′ such that $l' \geq l$.

6.3 Security Assumptions

Three main cryptographic tools we adopt here are hash function, chameleon hash function and digital signature. Therefore, the security of our system lies in the security of these cryptographic tools.

Hash functions are the key components for the blockchain construction, trans actions and micropayment. Our system relies on the underlying security of hash functions such as SHA256. The security requirement of hash functions is defined as follows:

DEFINITION 6.3 (Security of Hash Functions). The hash functions should meet the following security properties: onewayness and collision resistance. A brief description of these security properties has been given in Section 3 of this paper.

The construction of our micropayment scheme relies on the security of chameleon hash functions, whose security definition is given as follows:

DEFINITION 6.4 (Security of Chameleon Hash Functions). Given the public key and system parameters, it is computationally infeasible to find a collision. A brief description of the property of collision assistance is given in Section 3. Digital signatures are adopted in blockchain transactions and the construction of micropayment coins. We assume Existential Unforgeability of digital signa tures, which is the commonly accepted security standard of signature security. It has been defined as follows:

DEFINITION 6.5 (Security of Signatures). We assume the existential unforgeability of the signature scheme we select. Namely, there exists a signature forger, who cannot forge a signature on a challenge message m∗, which has never been queried in the signature query phase.

In blockchains, digital signatures are usually implemented in elliptical curves. For simplicity and convenience, we assume that they are operated in normal groups without loss of generality .

Volume: 16 Issue: 02 Pages: 6318– 6327 (2024) ISSN: 0975-0290

Apart from the cryptographic security assumptions, we also need to assume the underlying security of blockchain, in particular the integrity of ledgers. We define it below:

DEFINITION 6.6 (INTEGRITY OF LEDGERS). If the underling blockchain system is well functional, the integrity of ledgers is ensured in our system.

7 Micropayment Scheme

7.1 Basic Components
Here we present the basic components and requirements of our micropayment scheme.
• Users: Owner of the bitcoin U and recipient of a micropayment coin R. In addition, there is a group of miners who validate transactions in the blockchain.
• Setup of bitcoin owner: Select a pair of private and public keys (x, y) where $x \in Z^*_q$ and $y = g^x \text{mod} p$. Here, following the description above, we select a cyclic group Z^*_p of order q, where p = kq + 1 and p and q are large prime numbers, and select an element $g \in Z^*_p$. In addition, we require that (x, y) are properly selected for a digital signature scheme such as Schnorr signature or ElGamal signature, and can be used to construct a Chameleon hash function as well. The signature scheme will be used for transactions and micropayment.
• Digital signature: A digital signature scheme such as Schnorr signature or ElGamal signature defines a set of public and private keys associated with the signer, and usual operations of signing, denoted by $\sigma \leftarrow \text{Sign}_{sk}(m)$, where sk is the private signing key and m is the message. It takes as an input the private signing key sk and a message m and returns the signature σ on mes sage m. The verification is denoted by $\text{Verify}_{sk}(\sigma)$, which takes as an input a message, its signature σ and the corresponding public key pk and returns true or false, indicating validity or invalidity of the signature.
• Chameleon hash function: A chameleon hash function $\text{CHASH}_{pk_U}(m,r)$ defines a set of public and private keys associated with the owner U of the hash, denoted by pk_U and sk_U, respectively. A chameleon hash function takes as an input the public key pk_U, a message value m and a random string r and returns a hash value h.
• Verification of a chameleon hash: As a chameleon hash can be generated by anyone who has the public key, it cannot prove the ownership of the corresponding private trapdoor key. To prove the ownership of the private key, a zero-knowledge scheme can be applied. In this paper, we will not adopt any zero knowledge scheme. We will take advantage of our micropayment scheme to allow the owner of the private key to find a collision of the chameleon hash; since an efficient algorithm of collision can only be found when the private trapdoor key is known.

7.2 Generation of microcoins

Here we present the construction and transaction of the microcoins in the framework of bitcoin and blockchain. Suppose U and R are the owner of the microcoin and the receiver of the microcoin, respectively.

7.2.1 Assumed Structure of Cryptocurrency
Suppose there exists an underlying blockchain which supports transactions of the cryptocurrency. Although it is not necessary for bitcoins, for simplicity of representation, we select the bitcoin infrastructure, which is outlined as fol lows:
•A group of miners who help the validation of transactions.
•A group of users who act as senders and receivers relating to transactions. They hold their own public– private key pair. The public key is made available for the public.
•The sender signs the (hashed) public key of the recipient and the bitcoin to transfer it to the recipient.

7.2.2. The Micropayment Scheme
Following the algorithm definitions, here we present the detailed scheme for the micropayment.
$\text{KeyGen}(l^\lambda)$: Taking as an input a security parameter λ, select a pair of private and public keys (x, y) where $x \in Z^*_q$ and $y = g^x \text{mod} p$. Here, we select a cyclic group Z^*_p of order q where p = kq + 1 and p and q are large prime numbers and k is an integer, and select an element $g \in Z* p$. In addition, we require that (x, y) are properly selected for a digital signature scheme such as Schnorr signature or ElGamal signature, and can be used to construct a chameleon hash function as well. The signature scheme will be used for transactions and micropayment.
Sign(sk, m): Taking as an input the private key sk = x and a message m, where we can adopt the Schnorr signature algorithm, the algorithm returns a signature σ on m
$\text{Verify}(\sigma,pk,m)$: Taking as input a Schnorr signature σ, the corresponding public key pk = y and the corresponding message m, the algorithm returns true or false. (we have omitted the detailed algorithm)
CHashGen(pk, r, m) : The chameleon hash functions can be constructed from discrete log. This type of construction is based on the chameleon commitment scheme:

8 Security Analysis

(1) Our scheme is secure against Type I Adversary, if the underlining security of the hash function, chameleon hash function and digital signature is ensured.
(2)Our scheme is secure against Type II Adversary A. Given a recorded mi crocoin c^i, it is computationally infeasible for A to construct a new valid microcoin c^j.
(3)If the ring signature scheme RS = (Gen, Sign, Vrfy) is unforgeable and anonymous, the accountable assertion is extractable efficiently. Then, for any p.p.t. adversary A and security parameter λ, the micropayments scheme Π^m is secure as defined in Sect. 3.3.

9 Conclusion

We have proposed a new micropayment scheme for blockchain based micro payment transactions. Our scheme provides a number of features, including computational efficiency and security. It is for the time to build such type of microcoins in which we were able to construct a perfect hash chain by adopting chameleon hash functions. Our

Volume: 16 Issue: 02 Pages: 6318– 6327 (2024) ISSN: 0975-0290

approach can be used as a plugin to an existing crypto currency system with minimal changes.

In this paper, we analysed previous works, extracted the robustness requirements for achieving micropayments in decentralized blockchain-based system and explored efficient solutions to achieve these requirements.

References

[1]. Chaum, D., Fiat, A. and Naor, M. Untraceable Electronic Cash. Advances in Cryptology—CRYPTO '88, 8th Annu. Int. Cryptol. Conf., Santa Barbara, CA, USA, August 21–25, 1988, Proceedings Lecture Notes in Computer Science.

[2]. Chaum, D. (1982) Blind Signatures for Untraceable Payments. Advances in Cryptology: Proc. CRYPTO '82, Santa Barbara, CA, USA, August 23–25, 1982., pp. 199–203. Plenum Press, New York.

[3]. Nakamoto, S. (2008) Bitcoin: A Peer-to-Peer Electronic Cash System? https://bitcoin.org/bitcoin.pdf.

[4]. McCorry, P., M¨oser, M., Shahandashti, S.F. and Hao, F. (2016) Towards Bbitcoin Payment Networks. IACR Cryptology ePrint Archive, 2016, 408.

[5]. Micali, S. and Rivest, R.L. (2002) Micropayments Revisited. Topics in Cryptology—CT-RSA 2002, The Cryptographer's Track at the RSA Conference, 2002, San Jose, CA, USA, February 18–22, 2002, Proceedings, Lecture Notes in Computer Science, 2271, pp. 149–163. Springer.

[6]. Delmolino, K., Arnett, M., Kosba, A.E., Miller, A. and Shi, E. Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab. Financial Cryptography and Data Security—FC 2016 International Workshops, Lecture Notes in Computer Science, 9604 .

[7]. Burchert, C., Decker, C. and Wattenhofer, R. (2017) Scalable Funding of Bitcoin Micropayment Channel Networks— Regular Submission. Stabilization, Safety, and Security of Distributed Systems—19th Int. Sympos., SSS 2017, Boston, MA, USA, November 5–8, 2017, Proceedings, Lecture Notes in Computer Science, 10616, pp. 361–377. Springer..

[8]. Rivest, R.L. and Shamir, A. (1996) Payword and Micromint: Two Simple Micropayment Schemes. Security Protocols, International Workshop, Cambridge, United Kingdom, April 10–12, 1996, Proceedings, Lecture Notes in Computer Science, 1189, pp. 69–87. Springer.

[9]. Poon, J. and Dryja, T. (2016). The Bitcoin Lightning Network: Scalable off-Chain Instant Payments. https://lightning.network/ lightning-network paper.pdf.

[10]. Bitcoinj: Working with micropayment channels. (2013). https://bitcoinj.github.io/ working-with-micropayments.

[11]. PeterTodd: near-zero fee transactions with hub-and-spoke micropayments (2014).

[12]. Shelat, A., Shelat, A.: Micropayments for decentralized currencies. In: ACM SIGSAC Conference on Computer and Communications Security, pp. 207–218 (2015) .

[13]. Chiesa, A., Green, M., Liu, J., Miao, P., Miers, I., Mishra, P.: Decentralized anonymous micropayments. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017. LNCS, vol. 10211, pp. 609–642. Springer, Cham (2017). https://doi.org/10.1007/ 978-3-319-56614-6 21.

[14]. Pedersen, T.P.: Electronic payments of small amounts 24(495), 59–68 (1996).

[15]. Ruffing, T., Kate, A.: Liar, liar, coins on fire!: Penalizing equivocation by loss of bitcoins. In: ACM SIGSAC Conference on Computer and Communications Security, pp.

[16]. Nguyen, K.Q., Mu, Y. and Varadharajan, V. (1997) Microdigital Money for Electronic Commerce. 13th Annu. Comput. Secur. Appl. Conf. (ACSAC 1997), 8–12 December 1997, San Diego, CA, USA, pp. 2–8. IEEE Computer Society.

[17]. Rivest, R.L. (1997) Electronic Lottery Tickets as Micropayments. Financial Cryptography, First International Conference, FC '97, Anguilla, British West Indies, February 24–28, 1997, Proceedings, Lecture Notes in Computer Science, 1318, pp. 307–314. Springer.

[18]. Manasse, M.S. (1995) The Millicent Protocols for Electronic Commerce. First USENIX Workshop on Electronic Commerce, New York, New York, USA, July 11–12, 1995.

[19]. Lipton, R.J. and Ostrovsky, R. (1998) Micropayments via Efficient Coin Flipping. Financial Cryptography, Second International Conference, FC'98, Anguilla, British West Indies, February 23–25, 1998, Proceedings, Lecture Notes in Computer Science, 1465, pp. 1–15. Springer.

[20]. Heilman, E., Baldimtsi, F. and Goldberg, S. (2016) Blindly Signed Contracts: Anonymous On-Blockchain and Off Blockchain Bitcoin Transactions. IACR Cryptology ePrint Archive, 2016, 56.

[21]. Zhang, Y., Deng, R.H., Liu, X. and Zheng, D. (2018) Blockchain based efficient and robust fair payment for outsourcing services in cloud computing. Inf. Sci., 462, 262.

[22]. Glassman, S., Manasse, M., Abadi, M., Gauthier, P., Sobalvarro, P.: The millicent protocol for inexpensive electronic commerce (1995).

[23]. Baldimtsi, F., Chase, M., Fuchsbauer, G., Kohlweiss, M.: Anonymous transferable E-Cash. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 101–124. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46447-2 5.

[24]. Wheeler, D.: Transactions using bets. In: Lomas, M. (ed.) Security Protocols 1996. LNCS, vol. 1189, pp. 89–92. Springer, Heidelberg (1997).

[25]. Decker, C., Wattenhofer, R.: A fast and scalable payment network with bitcoin duplex micropayment channels. In: Pelc, A., Schwarzmann, A.A. (eds.)

Volume: 16 Issue: 02 Pages: 6318– 6327 (2024) ISSN: 0975-0290

SSS 2015. LNCS, vol. 9212, pp. 3–18. Springer, Cham (2015). https://doi.org/10.1007/978- 3-319-21741-3 1

[26]. Heilman, E., Baldimtsi, F., Goldberg, S.: Blindly signed contracts: anonymous onblockchain and off-blockchain bitcoin transactions. In: Clark, J., Meiklejohn, S., Ryan, P.Y.A., Wallach, D., Brenner, M., Rohloff, K. (eds.) FC 2016. LNCS, vol. 9604, pp. 43–60. Springer, Heidelberg (2016). https://doi.org/10.1007/978 3-662- 53357-4 4.

[27]. Ruffing, T., Kate, A.: Liar, liar, coins on fire!: Penalizing equivocation by loss of bitcoins. In: ACM SIGSAC Conference on Computer and Communications Security, pp. 219–230 (2015).

[28]. Bender, A., Katz, J., Morselli, R.: Ring signatures: stronger definitions, and constructions without random oracles. J. Cryptol. 22, 114–138 (2008) .

[29]. Ruffing, T., Kate, A.: Liar, liar, coins on fire!: Penalizing equivocation by loss of bitcoins. In: ACM SIGSAC Conference on Computer and Communications Security, pp. 219–230 (2015).

[30]. Ateniese, G. and de Medeiros, B. Identity-Based Chameleon Hash and Applications. Financial Cryptography, 8th International Conference, FC 2004, Key West, FL, USA, February 9-12, 2004. Revised Papers.

[31]. Ruffing, T., Kate, A.: Liar, liar, coins on fire!: Penalizing equivocation by loss of bitcoins. In: ACM SIGSAC Conference on Computer and Communications Security, pp. 219–230 (2015).

Biographies and Photographs

Gulab Das received the B.Sc. and M.Sc. degrees in Mathematics from Govt. N. PG College of Science, Raipur affiliated from Pt. Ravishankar Shukla University, Raipur, Chhattisgarh (India) in 2013 and 2015 respectively and also received Gold Medal in M.Sc. He is currently a research scholar at the Department of Mathematics in Govt. N. PG College of Science, Raipur. His main research interest include Security Tools on Blockchain Plateforms in the field of public key cryptography.

Dr. B. P. Tripathi is Assistant Professor in the Department of Mathematics in Govt. N. PG College of Science, Raipur affiliated from Pt. Ravishankar Shukla University, Raipur, Chhattisgarh(India). His field of are non-linear Analysis, fixed point theory and public key cryptography.

He has teaching experience of 28 years of undergraduate and postgraduate classes. He has written 2 books and published more than 45 research papers in various National and Internationa journals. His 5 scholars has awarded PhD degree and 3 scholars are pursuing his research work under the supervision of Dr. B.P. Tripathi. Two scholars are working on public key cryptography and 1 scholar is working on fixed point theory.