Real-time Money Routing by Trusting Strangers with your Funds

Martijn de Vos and Johan Pouwelse
Distributed Systems group, Delft University of Technology, The Netherlands
Email: m.a.devos-1@tudelft.nl

***Abstract*—We explore a new stage in the evolution of digital trust, trusting strangers with your funds. We address the trust issues when giving money to others and relying on them to forward it. For fraud identification, we leverage our deployed blockchain which gradually builds trust between interacting strangers. Our blockchain fabric, called *TrustChain*, records interactions between entities in a scalable manner. This work represents a small step towards a generic infrastructure for trust, moving beyond proven single vendor platforms like eBay, Uber and Airbnb.**

Expanding upon established trust relations, we designed, implemented and evaluated an overlay network: *Internet-of-Money*. Internet-of-Money routes money to different banks through individuals, so-called *money routers*. This removes the need for central banks, to handle a payment. Our network reduces the duration of traditional inter-bank payments from up to a day and even a few days during weekends, to mere seconds. Internet-of-Money is fully decentralized, scalable and privacy-preserving.

With real-world experimentations, we prove that Internet-of-Money enables fast money forwarding. We show that the overlay network is capable of discovering a majority of available money routers within a minute. Finally, we demonstrate how profit of cheating routers is limited and that misbehaviour is punished.

I. Introduction

Creating trust between strangers is at the core of numerous successful Internet companies. Starting 22 years ago, Craigslist offered an unmoderated mailing list of advertisements and gossip on which buyer and seller could be trusted. eBay formalised this in 1997 and introduced a star-based rating system that enables traders to build a trustworthy profile [1]. The e-commerce platform was launched at a time when people were still hesitant to use their credit card on a technology called The Internet. Nowadays, people let strangers sleep in their houses using Airbnb (since 2008). We trust Uber (since 2009) with our physical security and get into cars late at night with a driver that has never undergone a criminal background check or given a government license. These influential milestones in the evolution of digital trust are shown in Figure 1.

We continue this evolution of building trust. We created an operational platform for one of the most challenging and sensitive applications, having others handle your money.

Bitcoin created money without the need for banks [2]. In the past, people were required to trust a central bank and a host of other intermediaries when making payments [3]. The fundamental technology of Bitcoin, blockchain, radically reduced the need to trust financial middlemen. It bootstrapped an economy where no one can be stopped from spending their money. Despite widespread speculation and ecosystems being worth billions, blockchain in general suffers from scalability issues due to inefficient mechanisms for fraud prevention. Bitcoin is theoretically limited to seven transactions per second and Ethereum has a throughput of around 20 transactions per second [4]. Despite various scalability efforts like proof-of-stake and sharding, broader adoption of blockchain stays out.

While a majority of Internet users trust the company behind popular platforms, the events involving Mt. Gox highlighted how digital trust can be established and compromised [5]. Mt. Gox was at one point the largest Bitcoin exchange worldwide. In 2014, hackers stole Bitcoin, worth around $460 million at that time. This event, together with major data breaches in 2017 at high-profile companies like Uber and Equifax, exposed the weakness of centralized architectures [6]. They motivate research around decentralized technologies, like blockchain.

The generic problem of building trust between strangers resides on the edge of technology, sociology and behavioural science [7]. The question whether someone can be trusted, depends on properties like personality, level of authority, culture and past behaviour. In this research, we address the trust problem from a technological perspective, using tamper-proof interactions on a scalable blockchain. This structure is built to detect fraudulent behaviour and misrepresentation. We explore whether a trust model based merely on historical encounters is sufficient to trust strangers with your money.

With established trust relations, we demonstrate how one can transfer money within seconds between different banks by relying on others to act as financial intermediaries. In comparison to most proven platforms, our solution is designed to be fully decentralized and autonomous. Our work is motivated by slow money transfers to other banks using existing systems. Inter-banking payments often take up to a day or even a few days during weekends to arrive in the account of a beneficiary.

The main contributions of this work are as follows:

1) A trust model, based on repeated interactions and stored on a tamper-proof, scalable blockchain.
2) *Internet-of-Money*, a novel overlay network that allows real-time money routing to other banks.
3) Experimental quantification of the performance of our trust model, the speed of money transfers and the efficiency of our overlay network.
4) A framework to interface with multiple banks and to initiate payments to others using Internet-of-Money.

ISBN 978-3-903176-08-9 ©2018 IFIP

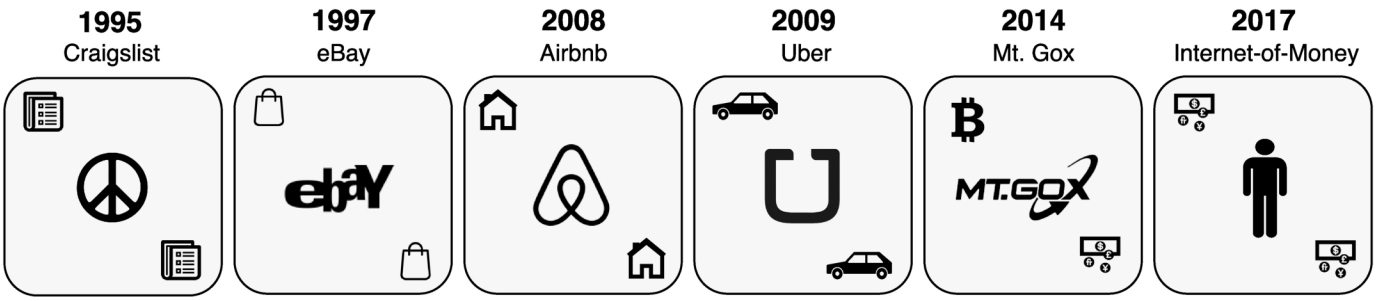


Fig. 1: Influential milestones in the evolution of digital trust.

II. Problem Description

Trust and fraud are essential problems to address when trusting others with your money. While most cryptocurrencies use a lottery system to stumble upon trustful executors, we rely on game theory to ensure honest behaviour has the largest rewards. We focus on the effective detection and punishment of fraudulent behaviour. While it is a common belief that money transfer systems should be safe against all kinds of fraud, we argue that it is sufficient for fraud to be detectable and punishable. This is comparable to the operation of credit card companies, which have to deal with a considerable amount of fraud on a daily basis. Detection of such fraud is non-trivial.

The trust problem in this work can be modelled by the prisoner's dilemma, where two entities can either cooperate or betray each other [8]. Betrayal is also called *defection*. In the iterative prisoner's dilemma, players cooperate or deflect iteratively and are able to punish opponents for their past decisions. We assume a *send and forward* model where a user first sends money to another user, who in turn forwards the money to someone else. Forwarding funds is considered cooperation whereas keeping the money is seen as defection. Detecting whether an entity has defected is a key requirement. Not cooperating should be punished by digital ostracism.

Many companies rely on centralized reputation mechanisms to manage trustworthiness of platform participants. In general, this leads to two problems. First, a solid track record built in one platform is often not reusable on other platforms. Second, building and maintaining an interaction history on multiple platforms simultaneously leads to fragmentation of one's trustworthiness scores. Users are increasingly being protected from such data silos by regulation [9]. Our aim is to devise a decentralized and generic reputation mechanism.

A mature research community exists around the design of decentralized reputation systems [10][11][12]. A notorious attack in decentralized systems occurs when a user first builds a high reputation by acting honest for some time and then abuses this accumulated trust for personal enrichment. This is also called the "pump and dump" method. Another challenging attack in decentralized networks is the Sybil Attack, where an individual creates multiple fake identities and initiate transactions with them to increase his or her standing in the community [13]. The Sybil Attack is hard to solve without trusted third parties, particularly in decentralized networks.

III. Settlement of Traditional Payments

Prior to elaborating how we can use trust and individuals to realise real-time money transfers, we briefly explore the process of performing a payment with existing infrastructure. International payment systems are often proprietary and lack transparency. The largest inter-bank communication network is SWIFT, the Society for Worldwide Interbank Financial Telecommunication [14]. In April 2017, SWIFT recorded an average of 28.38 million payments per day or around 328 per second. This legacy network was founded in the 1970s and programmed in a language from the 1950s (COBOL). While a majority of financial institutions worldwide rely on SWIFT, joining the network is an expensive and involved process. Due to the high costs when initiating cross-border payments, many users and companies are shut out of the system. It is estimated that back-office costs for international payments need to drop by 90% to 95% for banks to remain competitive [15].

The SWIFT network exposes high processing or *settlement* times for inter-bank payments, in particular for international payments. While many companies and banks are working on new platforms to enable instant (international) payments, there are various issues that should be addressed. These include real-time fraud detection and robust messaging standards [15].

A payment to another bank usually involves an intermediate settlement institution that is responsible for settling a payment between two parties [3]. This is often a central bank. The settlement institution acts as an intermediary in the payment chain and reduces settlement risks. Instead of handling a large number of payment instructions and settling them individually, settlement institutions usually aggregate outstanding payments and settle them all at once on predetermined times. This is called *net settlement* or *netting*.

While inter-bank payments take a considerable amount of time to settle, moving funds within the books of the same bank is significantly faster. This is called an *in-house payment*. In-house payments have a relatively low settlement duration, usually a few seconds, since no inter-bank communication is required.

IV. Our Money Routing Mechanism

Our mechanism to perform real-time payments is based on the observation that in-house payments are settled fast. For the banks that we tested, intra-bank money transfers are settled

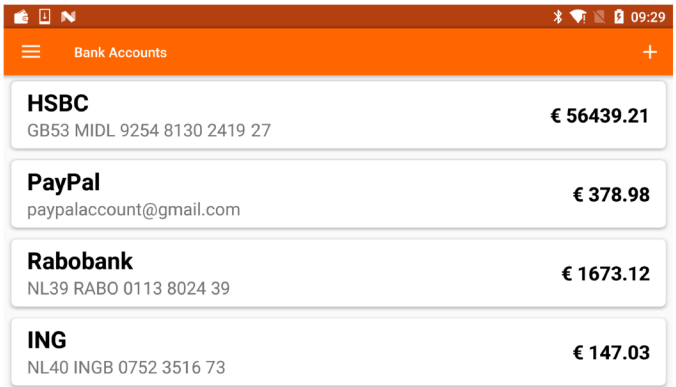


Fig. 2: Our Android application to interface with different banks and to route money in real-time.

within mere seconds (see Section VII-A). Instead of using a central bank as settlement institution, we build a network of individuals that have bank accounts with multiple banks to perform settlement on a gross basis. This works as follows: assume a Dutch buyer holding a Rabobank account, intends to pay a British merchant that holds a bank account with HSBC. When this buyer initiate a payment with existing software to the merchant, then the funds can take several days to arrive in the bank account of the merchant. However, when using an intermediary holding accounts both at Rabobank and HSBC, the buyer first sends the funds to the Rabobank account of this intermediary after which the buyer instructs the intermediary to forward the same amount of money from his HSBC account to the HSBC account of the merchant.

Since this way of sending money only involves two in-house payments, the merchant receives the funds within a few seconds. We call this process a *fast payment* We call the intermediary settling the transaction a *money router*. We use the terms *initiator* and *beneficiary* to indicate the initial sender and final receiver of a fast payment, respectively. A fast payment can be facilitated by multiple routers to increase efficiency and availability. Note that fast payments lead to mutations in the account balances of the involved money routers. This problem is addressed in Section VI.

Fast payments have three major advantages for users. First, it creates an open ecosystem for settlement activities, which benefits transparency and reduces the need for a central bank. Second, inter-bank settlement durations are significantly decreased, from days to seconds. Third, we reduce costs for inter-bank payments since no communication between banks is required, except when restoring balances (see Section VI).

Our system shares characteristics with services provided by Transferwise. Transferwise is a currency exchange service to offer a cheaper alternative to established institutions when making international payments. It routes payments not by transferring the sender's money directly to the recipient, but by redirecting them to the recipient of an equivalent transfer going in the opposite direction. The essential idea is to convert international money transfers into a sequence of local transactions. Their approach is comparable with our money router mechanism, as it also aims to reduce fees and improve efficiency of traditional payments. However, international payments with Transferwise can still take a few days to complete, depending on the settlement duration of involved banks.

In the remainder of this work, we elaborate our trust model and technical specifications of money routing. This includes an overlay network where any individual is able to quickly route money between bank accounts. A screenshot of our built Android application is shown in Figure 2[1]. The mobile application allows interfacing with different banks and the initiation of real-time payments using our overlay network.

V. Building Trust using Blockchain Constructs

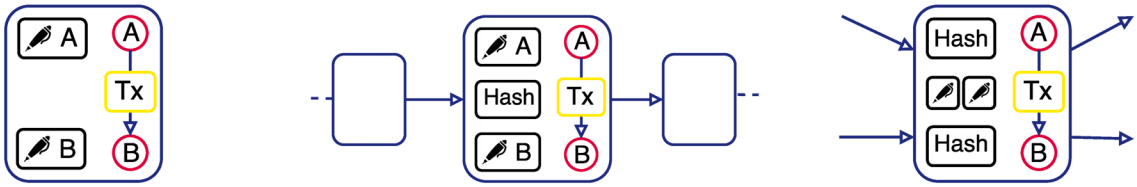
We now explain our deployed, scalable blockchain fabric to gradually build trust between fast payment initiators and money routers: *Trustchain*. Trustchain is designed around transacting entities and is able to accurately capture interactions between users. We have successfully explored usage of TrustChain for bandwidth accounting, attestations and decentralized trading in prior work [16]. For an elaborate evaluation of Trustchain, we refer the reader to our published article [17].

Figure 3 illustrates how a transaction is recorded between two users on Trustchain. Figure 3a shows a single transaction (Tx). Both parties sign the transaction with any cryptographically secure digital signature algorithm (our implementation uses ECDSA). This makes participation irrefutable and acts as an agreement for the transaction specifications. These digital signatures can efficiently be verified by others. After signing, the transaction is committed to the local databases of both transacting users.

A natural way to order records in a database is to chain them together, ordered by creation time. This is shown in Figure 3b where each record is extended with a pointer that points back to the prior record. In particular, this pointer is a hash computed from the description of the prior record using any cryptographically secure hashing algorithm (our implementation uses SHA256). Each record is equipped with a sequence number $s \in \mathbb{Z}$ (the sequence number of the genesis record is 1). This database organisation resembles a blockchain data structure. While cryptocurrencies like Bitcoin and Ethereum operate on a global blockchain, Trustchain gives each user their own personal chain.

Outside for the user operating a chain, the structure shown in Figure 3b is void of any control. Consequentially, a user is able to tamper with his historical transactions. For instance, individuals are able to remove transactions that are not beneficial for their standing in the network. After modification of a record, validity of the chain can simply be restored by recomputing all prior pointers. To protect against local modifications, we extend each record with an additional pointer that points to the prior record in the chain of the transaction counterparty. This ensures that each record has exactly two incoming and two outgoing pointers, as shown in Figure 3c.

[1]Android mobile application:http://www.ds.ewi.tudelft.nl/fileadmin/pds/homepages/vos/iom/iom.apk



(a) A transaction (Tx) between two users (A and B), with two digital signatures.

(b) A blockchain of transactions. Each record in the chain points back to the previous one.

(c) To increase security, each record also references a record in the chain of the other transaction participant.

Fig. 3: Recording a transaction between two users A and B in Trustchain.

When two users transact, their chains essentially become interleaved or "entangled". This property makes fraud impractical to hide since a counterparty is able to proof malicious activities by revealing his record of the disputed transaction. When users initiate more transactions with others, they quickly become entangled in the network, leading to a directed acyclic graph (DAG) structure as shown in Figure 4. This figure shows seven records, created by seven unique participants. Users are able to collect records stored by others. This ensures adequate replication of Trustchain records throughout the network.

Recording Payments: We use the Trustchain data structure to record money transfers between individuals. Each payment and fast payment is assigned a unique identifier. We define two different transaction types:

1) *commit*: This transaction is a public commitment by a money router to forward received funds. It is signed by the initiator of a fast payment and other money routers, prior to transferring any money. The transaction includes the identifier of a fast payment and account address of the money router that should forward funds.
2) *sent*: This transaction type is signed by two parties involved in an in-house payment and implies that money has been sent and received. This transaction includes the fast payment identifier and a boolean that is true if and only if the payment volume is above a threshold t.

We deliberately choose to hide the exact payment volumes due to privacy considerations, at the cost of reduced information.

Detect and Punish Fraud: The most challenging scenario occurs when a money router promises to forward money, but fails to do so. Since Trustchain provides us with a public ledger, disputes can be detected between transacting entities. Consider the situation when a router promised to forward money to Alice and claimed to have done so but Alice has not observed these funds (yet). Other individuals are informed of this situation when they observe a *commit* transaction, signed by both parties, and a *sent* transaction that is only signed by the money router that didn't forward the funds yet. As soon as such a dispute is detected, we do not consider this money router as intermediary for future fast payments until the dispute has been resolved. Note that a dispute can also occur when a router is unable to forward money, due to downtime of involved banks or insufficient account balance.

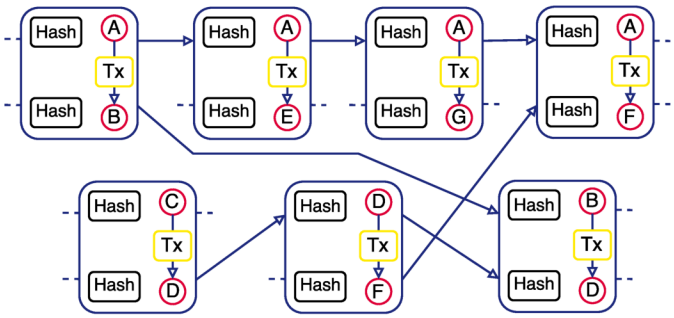


Fig. 4: The tamper-proof Trustchain data structure to record transactions.

In addition to the aforementioned scenario, a user can intentionally lie that he or she has not received funds from a money router. It is impossible to make statements about the status of a specific payment without access to both bank accounts involved in a payment. To resolve disputes, we propose to use input from a dispute arbitrator in the form of an official, digitally signed statement. The dispute arbitrator can be any company that is able to query bank accounts, for instance, the bank involved in a fast payment. A statement provides the status of a fast payment with a specific identifier and should be published on the Trustchain ledger. To discourage users from purposely creating disputes, the arbitrator should charge a small fee for publishing a statement, say €0.10. This fee should be covered by the party that made a false statement about money being sent or received. Note that dispute arbitration enables a new business model for banks.

Quantifying Trustworthiness: We now discuss a mechanism to quantify trustworthiness of honest money routers. Our proposed solution is based on past settlement services provided by money routers. We define a credit network G that models how much money a participant trusts to another individual. The graph is built using collected, dual-signed Trustchain transactions from others. Let $T_{a,b,R}$ indicate a successful money transfer from user a to b using the routers in the set R. Let (a, b, w) indicates a directed edge in G from user a to b with weight w. Now, each identity in our Trustchain network is modelled as a node in G. For each $T_{a,b,R}$ and each router $r \in R$, we create two directed edges: (a, r, w) and (b, r, w) where $w = min(0.01, t)$ (the minimum monetary value we trust to someone is €0.01). These edges represent trust in routers that have forwarded incoming money in the past.

To determine trust scores, we use an algorithm which has been studied extensively in related work, personalised PageRank [18]. The algorithm assigns a score between 0 and 1 to each node in G. These scores are used to pick intermediaries for money forwarding (see Section VI). We consider the node in G that performs the computation as trusted source. Using a reputation algorithm based on random walks is attractive due to its high scalability and low computational complexity. However, one might consider using a reputation algorithm based on maximum network flow to compute trust scores. In particular, we believe the Bazaar algorithm is suitable for this use case and provides additional security at the cost of increased computational requirements [19].

Preventing the Sybil Attack: We propose a mechanism called *router validation* to ensure that a specific bank account can only be operated by a single money router. The effectiveness of this method comes from the difficult and costly process of opening many accounts with different banks internationally. This addresses the challenging Sybil Attack, where an attacker operates multiple entities that use the same bank account for money routing. A router first registers a bank account by sending €0.01 to a trusted third party (TPP), for instance, a bank. The digital identity of TPPs are publicly available. TPPs sign and store a so-called *verify* transaction on Trustchain together with a money router when the payment is observed. This transaction uniquely connects a bank account to a money router. Routers reusing accounts across multiple identities can be identified by querying Trustchain records.

VI. System Design of Internet-of-Money

We expand upon fast payments and our trust model by designing a novel overlay network named *Internet-of-Money*. It operates on top of existing inter-bank payment systems, similar to how The Internet was built on top of the legacy telephone infrastructure.

The Money API: Except for the German FinTS payment protocol, there are no open standards yet for online banking. European legislation called PSD2 is forcing all EU banks to create open interfaces (APIs) [20]. We created one of the first open implementations capable of communicating with numerous banks. We combined banks in the Netherlands (Rabobank, ING and ABN Amro), the British bank HSBC and the Luxembourg payment provider PayPal [21][22]. We devised a single API to communicate with all these banks, called *The Money API*. The Money API provides primitives to login, fetch account balance, query mutations, initiate payments to other accounts and register devices. This library is designed to be extendible and we have partial support for banks in Italy, Greece, Sri Lanka, Turkey and Germany. Our open source[2] library is currently being tested.

Money Routers: Each money routers must offer settlement services with at least two different bank accounts. Having many money routers in the network directly benefits availability and load balancing. A study conducted by NGData indicated that 37.7% of the respondents held accounts at different banks and are able to act as settlement intermediary for money transfers [23]. To create incentives for users to operate a money router, we include transaction fees. Transaction fees can be either fixed, defaulting to €0.01, or a percentage of a fast payment volume. These fees are necessary to cover costs enforced by banks when initiating cross-border payments or when using business accounts to route money. In addition, users can specify a minimum account balance to avoid taking costs when their balance becomes negative. In the remainder of this work, we assume transaction fees are fixed. We also consider an analysis of monetary incentives out of scope and not fundamental for the prototype evaluated in this work.

Note that our design also allows the role of money router to be fulfilled by a single trusted third party or by a few selected trustworthy entities (i.e. financial institutions). A more centralized architecture would mitigate some of the trust and security issues that arise from full decentralization. However, we consider open enrollment (the opportunity for any user to act as a money router) a cardinal property of our system.

Router Discovery: We designed a gossip protocol for discovery of available money routers, based on utility. Like all our proposed infrastructure, it does not depend on any server, company, or other central entity. If Alice wishes to discover a new router, she asks one of her known peers, say Bob, to introduce a router to her. Now, Bob tries to introduce a router to Alice through which she can route money. In general, the algorithm prioritizes routers that provide the most benefit to Alice. If Bob has no router in his set of known peers that are able to provide new services to Alice, he will introduce a random router to Alice. Repeating this gossiping protocol quickly converges to a network with connections between individuals able to provide routing services for each other. An evaluation of this mechanism is given in Section VII-B.

Building a Money Circuit: Prior to transferring money, an initiator of a fast payment starts by selecting eligible routers that are capable of handling the upcoming fast payment. We define a *money circuit* as the set of peers that are involved in a fast payment. This set contains at least one initiator and one beneficiary, and optionally one or more money routers. A money circuit that contains n money routers, is called a n-hop circuit. Building a money circuit proceeds in a depth-first manner and starts with the initiator selecting a router, say r, that is capable of routing money to another account. Next, the initiator sends an *extend* message to r which contains the payment volume and the destination bank account of the fast payment. r responds with a boolean that indicates whether r has sufficient funds to handle the transfer. The response also includes a list of routers that are able to extend the money circuit, and the transaction fee charged by r. If r is able to handle the transfer, the initiator picks a router to extend the circuit with and sends an *extend* message again. These routers are picked based on trustworthiness scores. This process repeats until the initiator built a money circuit that

[2]The Money API source code:
http://www.ds.ewi.tudelft.nl/fileadmin/pds/homepages/vos/iom/internet_of_money.zip

can handle the fast payment. Users are able to change the maximum number of routers in a circuit, which defaults to 3.

The trust model discussed in Section V is based purely on past transactions. It is useful to consider other properties when picking eligible money routers, such as transaction fees, availability, reliability or network latency. Depending on the situation, one might favour low network latency or competitive transaction fees over trustworthiness.

Transferring Money: We now elaborate the process of transferring money over a n-hop circuit. If $n = 0$, money is sent directly to the beneficiary using exactly one in-house payment and no money routers. A single *sent* transaction is created between the fast payment initiator and beneficiary.

When a money circuit involves one or more money routers ($n \geq 1$), the fast payment is facilitated by intermediaries. Let r_i indicate the i-th router in the circuit (r_1 represents the first router). The initiator starts by sending a message to r_1, containing the payment volume and all subsequent routers involved in the money circuit, including the final beneficiary of the fast payment. Next, the initiator initiates a *commit* transaction with r_1 and sends the money. r_1 now starts to poll for the money and finally constructs a *sent* transaction when funds are observed. r_1 forwards the funds to the next router or the beneficiary and this process repeats until the money arrives in the bank account of the beneficiary. The final transfer to the beneficiary does only result in a *sent* transaction. Thus, a fast payment with n intermediaries results in $2n + 1$ new records.

Risk Mitigation: In addition to our trust model, we propose two risk mitigation techniques to reduce counterparty risk when using money routers:

1) *Incremental settlement*: A key risk mitigation technique is to avoid making a single, large payment at once. Instead, a payment is divided into n smaller inter-bank payments. While this increases duration of a fast payment by a factor n, it significantly reduces risk and incentives for intermediaries to compromise money. We believe that reduced risk for some increased latency is a desirable trade-off in Internet-of-Money.
2) *Multi-flow payments*: We uniformly divide a fast payment amongst multiple, distinct money circuits. This results in smaller payments through intermediaries.

While these individual strategies are viable to mitigate counterparty risk, combining them results in a significant reduction of the value at stake, at the cost of additional latency and communication overhead. We evaluate the effectiveness of these strategies in Section VII-B.

Router Recharging: Since funds arrive in one account and leave another, money routers might become insolvent at one point in time, unable to route additional funds. This can be addressed by handling fast payments going in the opposite direction, which restores account balances. However, initiation of these fast payments is outside the control of money routers. Balances can also be restored by initiating a payment from the account with excessive balance to the other bank account. Since this involves an inter-bank payment, settlement might be slow and in turn, this negatively impacts router availability.

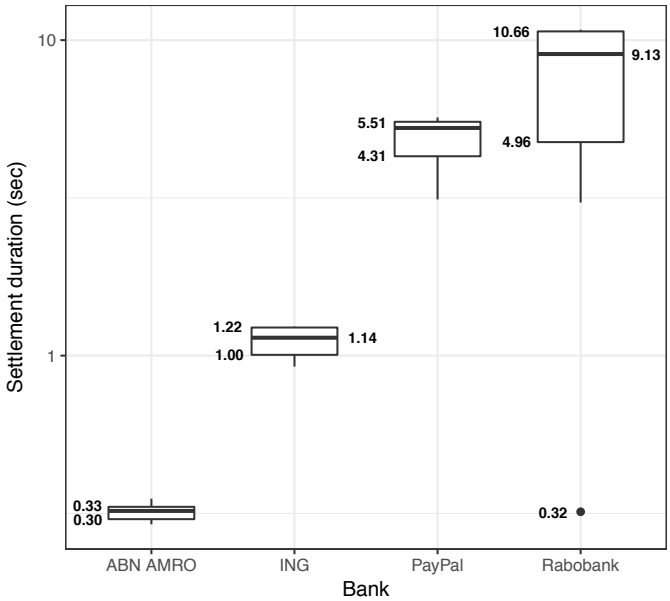


Fig. 5: Settlement durations of in-house payments for four supported banks.

We envision an infrastructure where routers help each other to restore balances, effectively creating a two-sided market with capacity supply and demand. For instance, a router can offer PayPal capacity in return for HSBC funds. While this is an efficient method to restore balances, only requiring in-house payments, we consider the design and implementation of such a mechanism as future work.

VII. Experiments and Evaluation

We now evaluate the performance of money routers, speed of router discovery within Internet-of-Money and the effectiveness of our trust model.

A. Performance of Money Routing

This section concentrates on the performance of fast payments using money routers. All these experiments are conducted with real bank accounts and real money.

Settlement duration of in-house payments: To determine settlement duration of in-house payments for each bank, we send €0.01 ten times between two accounts with different holders, within the same bank. By adding a unique identifier to the description field of a payment, we are able to track payments and accurately measure settlement times. The experiment is executed with two clients on two different computers, with a polling interval of 500 milliseconds, to avoid hammering the bank servers. Polling starts when the payment request has been finished by the sending party. The results are shown in Figure 5, with a non-linear vertical axis. Only one bank, ABN AMRO, has sub-second settlement times with an average duration of 320 milliseconds. ING is slower with 1109 milliseconds on average. PayPal and Rabobank show settlement durations that are an order of magnitude slower, averaging to 4.82 and 7.61 seconds respectively. When performing measurements for the

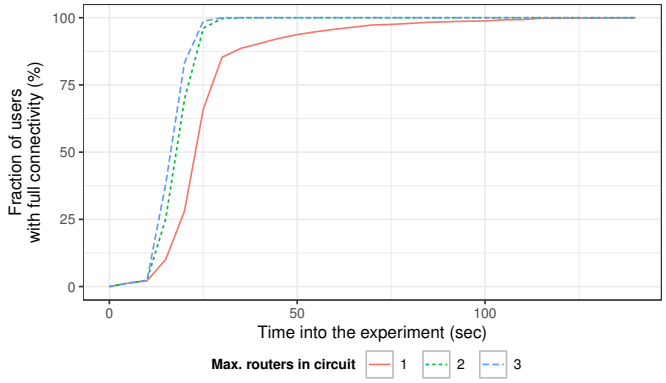


Fig. 6: Performance of router discovery under a varying number of maximum hops in a money circuit.

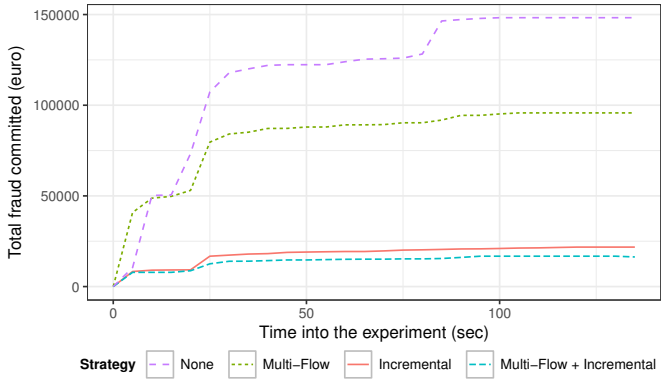


Fig. 7: The effectiveness of fraud prevention, with different risk mitigation strategies.

Rabobank, we observed a notable outlier with a settlement time of 320 milliseconds. This observation can be explained if we assume that similar internal payments might be handled in different ways by the Rabobank. This experiment demonstrates that in-house payments are usually settled within seconds.

International Real-time Money Routing: Next, we focus on the performance of an international fast payment and measure the duration of a money transfer from Rabobank to ABN AMRO, using two money routers. This experiment aims to show the viability and speed of Internet-of-Money. Figure 8 shows the experimental setup and timeline of our experiment.

First, an initiator sends funds from his or her Rabobank account to the first router (holding an account at Rabobank and PayPal), and informs it about the sent funds. Next, the first router starts polling for incoming funds, with an interval of 500 milliseconds. When the first router observes the funds, it forwards them to the second router (holding an account at PayPal and ABN AMRO) and informs this router. When the second router observes the funds, it forwards the money from it's ABN AMRO account to the ABN AMRO account of the beneficiary. In total, three in-house payments are made, with six different bank accounts.

From Figure 8, we conclude that it takes 15.85 seconds in total for money to arrive in the bank account of a beneficiary when using two intermediate routers. A significant amount of time is spent on waiting for the funds to arrive in the PayPal account of the second router, around 6 seconds or 38% of the total duration. The average time to perform a payment is 2.14 seconds and initiation of payments take 41% of the total duration. The average time that a transaction is in transit is 3.02 seconds. The total time to perform a fast payment is heavily influenced by the type and number of intermediate routers. This experiment demonstrates that Internet-of-Money is capable of real-time money routing to other banks.

B. Overlay Evaluation

The purpose of the following experiments is to quantify the performance of our money router overlay. This includes an evaluation of our trust model and effectiveness of fraud detection. We implemented our trust model and Internet-of-Money overlay network in the Python programming language. Our implementation is built upon the Dispersy framework, providing primitives for peer discovery, decentralized communication and secure messaging [24].

Experimental Setup: The following real-world emulations are executed on the DAS-5 supercomputer, using 50 instances per node [25]. We deploy our experiment using the Gumby framework and we create a scenario file where we schedule actions at specific times. All code used during these experiments is open source[3]. Due to the limited number of accounts we own and to avoid a large load on the banking infrastructure, simulated accounts are used during this experiment. We assume a total of five different banks and devised a basic RESTful banking server that handles account creation, payments, balance queries and mutation requests. Distribution of bank accounts amongst users follows the data as published in the NGData customer banking survey (we assume that every user owns at least one bank account) [23].

Router Discovery: We evaluate the efficiency of the router discovery protocol discussed in Section VI. During the experiment, we record the connected peers for each user at a fixed interval (every 5 seconds). We determine whether this user is capable of transferring money to all five different bank accounts, using at most one, two and three intermediate money routers respectively.

Figure 6 shows the performance of router discovery in the Internet-of-Money overlay. The horizontal axis denotes the time into the experiment. The vertical axis indicates the percentage of users that are able to make fast payment to all five banks, or are fully connected. We vary the maximum number of routers in a money circuit. As expected, it takes longer before users are able to build circuits to all other banks using only one router, compared to three routers. However, the differences are marginal. In general, router discovery happens fast: 50% of all users are able to make fast payments to all banks within 25 seconds after the experiment starts. 40 seconds into the experiment, this percentage increased to 90%. Note

[3]https://github.com/devos50/gumby/tree/iom_experiment

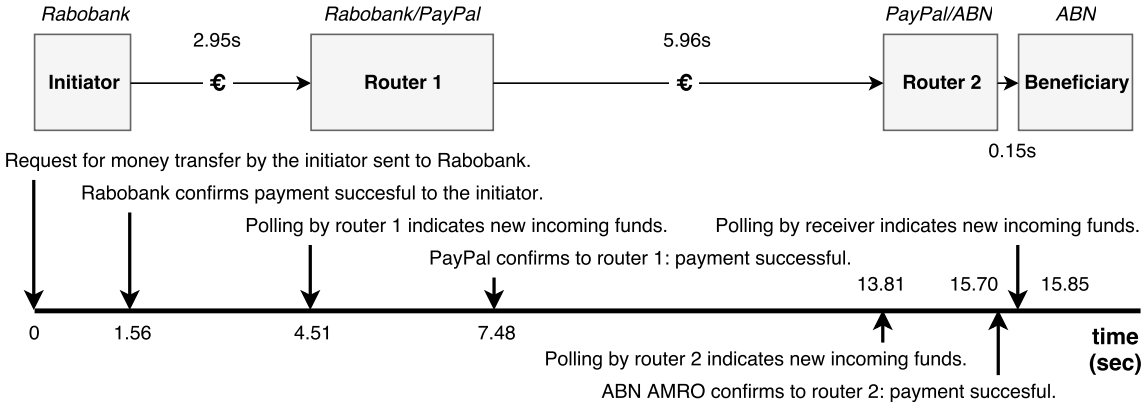


Fig. 8: Timeline of an international fast payment from Rabobank to ABN AMRO, using two money routers.

that it takes longer before *all* users are fully connected using at most one intermediate router: 140 seconds.

Fraud Detection: Our final experiment focusses on the effectiveness of fraud detection (see Section V). To this end, we emulated 200 users with one or more bank accounts. Every five seconds, each user with a single account initiates a fast payment to another entity that has exactly one account of a different type. This forces a money router in the established circuits. The volume of each fast payment is picked from a uniform random distribution between €0.01 and €1000. We challenge ourselves and assume that every user with at least two different bank accounts is malicious and has a 50% probability of committing fraud and not forwarding received funds during a fast payment. To improve router availability, we connect all peers together before the experiment starts. In total, we schedule payments which volume sums to €1,251,848.35.

The results are shown in Figure 7. The horizontal axis denotes the time into the experiment in seconds, after users start performing fast payments to each other. The vertical axis shows the total amount of committed fraud in Euro. We run the experiment four times with different risk mitigation strategies, namely incremental settlement (we split each fast payment in five equal parts) and/or multi-flow payments. The figure hints that the amount of fraud is capped and that malicious routers are successfully excluded from money circuits. Without any risk migration strategy, malicious routers are able to steal €1,544 on average during the whole experiment, indicating that fraudulent routers are able to commit fraud multiple times. This can be addressed to the fact that they are included in multiple money circuits roughly at the same time. If we consider risk mitigation strategies, we see that the combination of multi-flow payments and incremental settlement leads to the lowest amount of fraud possible, on average €174. Using exclusively incremental settlement leads to a slightly higher amount of fraud.

VIII. Discussion

We now discuss this research from various perspectives.

Legal: The idea of directly sharing funds with others, without a central bank involved, challenges existing regulation. Routing money through other bank accounts resembles activity performed by financial settlement institutions and might require a legal prerequisite in the form of a banking license. The PSD2 regulation states that trusted third parties (TPPs) can be authorized by end-users to perform financial activities on their behalf [20]. However, it is unclear whether the definition of a TPP includes money routers. Another consideration is responsibility when a mistaken payment is initiated. Finally, compatibility of our system with (inter)national anti-money laundry regulations is uncertain. Exploring legal compliance of this work is a fundamental requirement for broader adoption.

Limitations: While we have proven the viability of our idea, there are several limitations that must be addressed prior to broader adoption. We noticed that banks are not used to our dynamic way of initiating money transfers and our accounts got blocked several times due to suspected fraudulent behaviour. An open ecosystem for settlement demands changes by banks and it is an open question whether they are willing to do so. On the other hand, many banks are already forced to innovate their legacy systems to remain competitive [15].

Additionally, we observed that some banks require two-factor authentication when transferring funds to unknown bank accounts. This limits automation of money transfers since a manual action by the user is required for a payment to proceed.

Privacy: We consider privacy an important requirement of our open platform and expose minimal information about money flows. The current privacy model in Internet-of-Money is effective but open for extension. Decentralized path-based transaction networks, for instance, SpeedyMurmurs, aim to solve this specific problem [26].

Scalability: Our overlay network is scalable, due to the absence of global consensus. However, techniques like incremental settlement lead to additional payments and a higher load on the banks. In addition, the choice of reputation mechanism used in Internet-of-Money influences scalability.

IX. Related Work

The last few years, there has been a steep increase in Fintech start-ups, eager to disrupt existing financial services. Hawala is an informal system to transfer value, without actually moving

money [27]. It consists of a network of hawala brokers, that take a small commission. In contrast to our system, trust in hawala is cultivated in an analogue manner whereas our model depends on a digital solution.

Innovation in the financial sector has been catalysed by the popularity of Blockchain technology, aiming to build trust between strangers without involvement of centralized authorities. Bitcoin has proven that a sustainable currency can be built without a central bank in control [2]. However, wide-spread adoption stays out due to its volatile pricing, high transaction fees, relatively slow confirmation times and unsure future. The Lightning Network aims to improve scalability of Bitcoin by providing bi-directional payment channels between users [28]. Payments between two users not directly connected with a payment channel, are realised by routing payments through channels of other users. This has similarities with money routing in Internet-of-Money. New usages of blockchain technology are focussed around the way users transfer money and other assets. The Ripple project, supported by various major banks, attempts to build a connected network of financial institutions and payment providers [29]. Their solution aims to significantly speed up traditional money transfers, lower costs and provide support for high-volume transactions. R3 Corda can be compared to Trustchain since they share the idea that a ledger with global consistency is often not necessary [30].

While blockchain solutions are slowly being adopted, the aforementioned systems all aim to increase utility by building a financial network from scratch. In comparison, Internet-of-Money is built upon existing, proven infrastructure, making migration towards our system effortless.

X. Conclusions and Future Work

We explored a new stage in the evolution of digital trust and addressed the problem of trusting strangers with your money. The tamper-proof Trustchain structure provides a scalable and public trace of historical interactions, and allows detection and punishment of potential fraud. We expand upon this with an overlay network to transfer money within seconds to others, using other network participants as financial intermediaries. This mechanism depends on the fast settlement of in-house payments. Our open ecosystem dramatically improves speed when initiating cross-border payments while preserving privacy and scalability. Our experiments demonstrated the efficiency of in-house payments and effectiveness of money routers. Additionally, we have proven that our fraud detection mechanism, together with incremental settlement and multi-flow payments, limits misuse and punishes malicious behaviour. However, there are various legal issues and limitations that should be addressed, mostly by financial institutions, before broader usage can be realised.

This work is an important milestone in our ambitious vision to create the *programmable economy*. Ongoing work towards this goal addresses self-sovereign identity, scalable blockchain consensus compatible with Trustchain, and decentralized marketplaces. We refer the interested reader to our scientific overview article [16].

Acknowledgements

The authors would like to thank Laurens Versluis for his initial contributions to the design and implementation of The Internet-of-Money.

References

[1] P. Resnick *et al.*, "Trust among strangers in internet transactions: Empirical analysis of ebays reputation system," *The Economics of the Internet and E-commerce*, vol. 11, no. 2, pp. 23–25, 2002.

[2] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[3] T. Kokkola, *The payment system: Payments, securities and derivatives, and the role of the Eurosystem*. European Central Bank, 2011.

[4] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work vs. bft replication," in *International Workshop on Open Problems in Network Security*. Springer, 2015, pp. 112–125.

[5] R. McMillan, "The inside story of mt. gox, bitcoins 460 million disaster," *Wired. March*, vol. 3, 2014.

[6] "The uber data breach has implications for us all." [Online]. Available: https://www.ft.com/content/e2bf6caa-d2cb-11e7-a303-9060cb1e5f44

[7] Z. Yan and S. Holtmanns, "Trust modeling and management: from social trust to digital trust," *IGI Global*, pp. 290–323, 2008.

[8] D. M. Kreps *et al.*, "Rational cooperation in the finitely repeated prisoners' dilemma," *Journal of Economic theory*, vol. 27, no. 2, pp. 245–252, 1982.

[9] B.-J. Koops, "The trouble with european data protection law," *International Data Privacy Law*, vol. 4, no. 4, pp. 250–261, 2014.

[10] R. Delaviz *et al.*, "Sybilres: A sybil-resilient flow-based decentralized reputation mechanism," in *ICDCS, 2012*. IEEE, 2012, pp. 203–213.

[11] S. D. Kamvar *et al.*, "The eigentrust algorithm for reputation management in p2p networks," in *WWW*. ACM, 2003, pp. 640–651.

[12] M. Srivatsa *et al.*, "Trustguard: countering vulnerabilities in reputation management for decentralized overlay networks," in *Proceedings of WWW'05*. ACM, 2005, pp. 422–431.

[13] J. R. Douceur, "The sybil attack," in *International Workshop on Peer-to-Peer Systems*. Springer, 2002, pp. 251–260.

[14] *SWIFT payment system*. [Online]. Available: https://www.swift.com

[15] McKinsey, *Global Payments 2016*, 2016 (accessed November 27, 2017).

[16] J. Pouwelse *et al.*, "Laws for creating trust in the blockchain age," *European Property Law Journal*, 2017.

[17] P. Otte *et al.*, "Trustchain: A sybil-resistant scalable blockchain," *Future Generation Computer Systems*, 2017.

[18] L. Page *et al.*, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[19] A. Post *et al.*, "Bazaar: Strengthening user reputations in online marketplaces," in *Proceedings of NSDI11*, 2011, p. 183.

[20] M. Cortet *et al.*, "Psd2: The digital transformation accelerator for banks," *Journal of Payments Strategy & Systems*, vol. 10, no. 1, pp. 13–27, 2016.

[21] J. Doe and J. Pouwelse, "A vulnerability analyis of smartphone banking applications," 2015, unpublished research.

[22] J. Awesome and J. Pouwelse, "A vulnerability analysis of mobile banking applications," 2015, unpublished research.

[23] N. n, *NGDATA 2014 Consumer Banking Survey*, 2014 (accessed November 20, 2017). [Online]. Available: http://www.ngdata.com/wp-content/uploads/NGDATA-2014-consumer-banking-survey-brief.pdf

[24] N. Zeilemaker, B. Schoon, and J. Pouwelse, "Dispersy bundle synchronization," *TU Delft, Parallel and Distributed Systems*, 2013.

[25] H. Bal *et al.*, "A medium-scale distributed system for computer science research: Infrastructure for the long term," *Computer*, vol. 49, no. 5, pp. 54–63, 2016.

[26] S. Roos, P. Moreno-Sanchez, A. Kate, and I. Goldberg, "Settling payments fast and private: Efficient decentralized routing for path-based transactions," *arXiv preprint arXiv:1709.05748*, 2017.

[27] P. M. Jost and H. S. Sandhu, *The hawala alternative remittance system and its role in money laundering*. Interpol, 2003.

[28] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments. 2016," 2015.

[29] D. Schwartz *et al.*, "The ripple protocol consensus algorithm," *Ripple Labs Inc White Paper*, vol. 5, 2014.

[30] R. G. Brown, "Introducing r3 corda: A distributed ledger for financial services," *R3, April*, vol. 5, 2016.