Ripple: Overview and Outlook

Frederik Armknecht[1], Ghassan O. Karame[2(✉)], Avikarsha Mandal[3], Franck Youssef[2], and Erik Zenner[3]

[1] University of Mannheim, Mannheim, Germany
armknecht@uni-mannheim.de
[2] NEC Laboratories Europe, 69115 Heidelberg, Germany
{ghassan.karame,franck.youssef}@neclab.eu
[3] University of Applied Sciences, Offenburg, Germany
{avikarsha.mandal,erik.zenner}@hs-offenburg.de

Abstract. Ripple is a payment system and a digital currency which evolved completely independently of Bitcoin. Although Ripple holds the second highest market cap after Bitcoin, there are surprisingly no studies which analyze the provisions of Ripple.

In this paper, we study the current deployment of the Ripple payment system. For that purpose, we overview the Ripple protocol and outline its security and privacy provisions in relation to the Bitcoin system. We also discuss the consensus protocol of Ripple. Contrary to the statement of the Ripple designers, we show that the current choice of parameters does not prevent the occurrence of forks in the system. To remedy this problem, we give a necessary and sufficient condition to prevent any fork in the system. Finally, we analyze the current usage patterns and trade dynamics in Ripple by extracting information from the Ripple global ledger. As far as we are aware, this is the first contribution which sheds light on the current deployment of the Ripple system.

Keywords: Ripple · Bitcoin · Security · Forks

1 Introduction

The wide success of Bitcoin has lead to a surge of a large number of alternative crypto-currencies. These include Litecoin [1], Namecoin [2], Ripple [6,38], among others. Most of these currencies are built atop the Bitcoin blockchain, and try to address some of the shortcomings of Bitcoin. For example, Namecoin offers the ability to store data within Bitcoin's blockchain in order to realize a decentralized open source information registration based on Bitcoin, while Litecoin primarily differs from Bitcoin by having a smaller block generation time, and a larger number of coinbases, etc. While most of these digital currencies are based on Bitcoin, Ripple has evolved almost completely independently of Bitcoin (and of its various forks). Currently, Ripple holds the second highest market cap after Bitcoin [4]. This corresponds to almost 20 % of the market cap held by Bitcoin. Recently, Ripple Labs have additionally finalized the financing of an

M. Conti et al. (Eds.): TRUST 2015, LNCS 9229, pp. 163–180, 2015.
DOI: 10.1007/978-3-319-22846-4_10

additional 30 million USD funding round to support the growth and development of Ripple [5].

Ripple does not only offer an alternative currency, *XRP*, but also promises to facilitate the exchange between currencies within its network. Although Ripple is built upon an open source decentralized consensus protocol, the current deployment of Ripple is solely managed by Ripple Labs. Originally, the Ripple network was created with a limited supply of 100 billion XRP units; 20 % of those units are retained by Ripple founders, 25 % are held by Ripple Labs, while the remaining 55 % are set to be sold. This represents the largest holdback of any crypto-currency [4], but has not apparently stopped the adoption of Ripple by a considerable fraction of users. At the time of writing, Ripple claims to have a total network value of approximately 960 million USD with an average of almost 170 accounts created per day since the launch of the system [33]. Moreover, there are currently a number of businesses that are built around the Ripple system [14,20]. For instance, the International Ripple Business Association currently deploys a handful of Ripple gateways [22], market makers [23], exchangers [21], and merchants [24] located around the globe.

Although crypto-currencies are receiving considerable attention in the literature [11,16,28,31,37], there are surprisingly no studies—as far as we are aware—that investigate the Ripple system. In this paper, we remedy this problem and we analyze the deployment and security provisions of the Ripple payment system. More specifically, we overview the Ripple protocol and discuss the basic differences between the current deployments of Ripple and Bitcoin. Motivated by recent forks in the Ripple consensus protocol [25], we provide a new necessary and sufficient condition that provably prevent the realization of a fork in Ripple. Finally, we extract information on the current usage patterns and trade dynamics in Ripple from almost 4.5 million ledgers which were generated in the period between January 2013, and January 2015. Our findings suggest that—although it has been introduced almost 2 years ago—most Ripple users seem inactive and their trade volume is not increasing. As far as we are aware, this is the first contribution which investigates the current deployment of Ripple.

The remainder of this paper is structured as follows. In Sect. 2, we detail the Ripple protocol and the underlying consensus protocol. We also discuss the security and privacy provisions of Ripple in relation to the Bitcoin system. In Sect. 3, we analyze the conditions for forking in Ripple. In Sect. 4, we analyze the current usage patterns of Ripple by extracting information from the Ripple ledgers. In Sect. 5, we discuss related work in the area, and we conclude the paper in Sect. 6.

2 The Ripple Protocol

In what follows, we introduce and detail the Ripple system. We also analyze Ripple's consensus protocol and compare it to Bitcoin.

2.1 Overview of Ripple

Ripple [38] is a decentralized payment system based on credit networks [19,29]. The Ripple code is open source and available for the public; this means that anyone can deploy a Ripple instance. Nodes can take up to three different roles in Ripples: *users* which make/receive payments, *market makers* which act as trade enablers in the system, and *validating servers* which execute Ripple's *consensus* protocol in order to check and validate all transactions taking place in the system.

Ripple users are referenced by means of pseudonyms. Users are equipped with a public/private key pair; when a user wishes to send a payment to another user, it cryptographically signs the transfer of money denominated in Ripple's own currency, XRP, or using any other currency. For payments made in non-XRP currencies, Ripple has no way to enforce payments, and only records the amounts owed by one entity to the other. More specifically, in this case, Ripple implements a distributed credit network system.

A non-XRP payment from A to B is only possible if B is willing to accept an "I Owe You" (IOU) transaction from A, i.e., B trusts A and gives enough credit to A. Hence, A can only make a successful IOU payment to B if the payment value falls within the credit balance allocated by B to A. This may be the case, e.g., if the participants know each other, or if the involved amounts are rather marginal; typically however, such transactions require the involvement of "market makers" who act as intermediaries. In this case, enough credit should be available throughout the payment path for a successful payment.

For example, a trust line can be established between market maker $U1$ and A (cf. Fig. 1) by A depositing an amount at $U1$. In our example, A wants to issue a payment to B with the amount of 100 USD. Here, the payment is routed from $A \rightarrow U1 \rightarrow U2 \rightarrow U4 \rightarrow B$. This is possible because available credit lines are larger than the actual payment for every atomic transactions. Notice that we did not route through $U3$ as there is not enough credit available between $U1 \rightarrow U3$. However, we note that it is possible to break down the payment amount at $U1$, route a payment below 90 USD through $U1 \rightarrow U3 \rightarrow B$ and transfer the rest through $U1 \rightarrow U2 \rightarrow U4 \rightarrow B$ (extra fee at $U3$ required). In typical cases, Ripple relies on a path finding algorithm which finds the most suitable payment path from the source to the destination. By implementing credit networks, Ripple can act as an exchange/trade medium between currencies; in case of currency pairs that are traded rarely, XRP can act as a bridge between such currencies.

Ripple's Ledger: Ripple maintains a distributed ledger which keeps track of all the exchanged transactions in the system. Ledgers are created every few seconds, and contain a list of transactions to which the majority of *validating servers* have agreed to. This is achieved by means of Ripple's consensus protocol [38] which is executed amongst validating servers. A Ripple ledger consists of the following information: *(i)* a set of transactions, *(ii)* account-related information such as account settings, total balance, trust relation, *(ii)* a timestamp, *(iv)* a ledger number, and *(v)* a status bit indicating whether the ledger is validated or not. The most recent validated ledger is referred to as the *last closed ledger*. On the other hand, if the ledger is not validated yet, the ledger is deemed *open*.

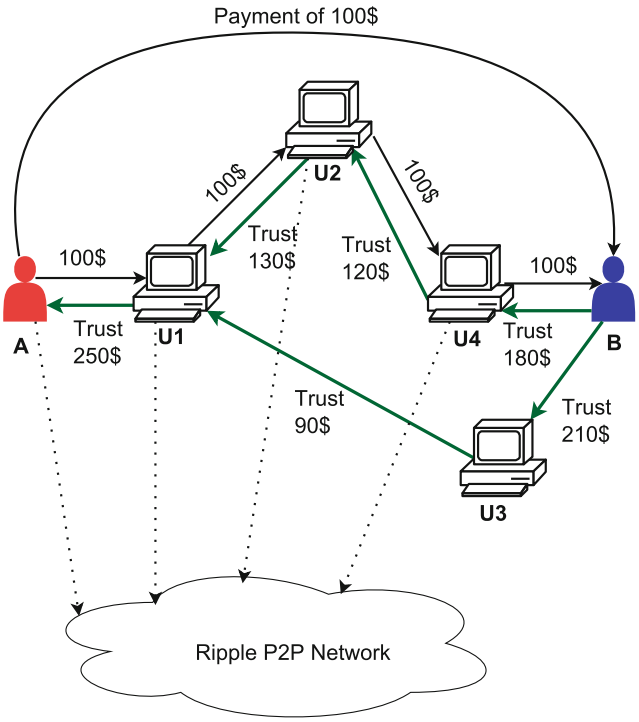


Fig. 1. Exemplary sketch of IOU payments in Ripple. Here, A wants to pay 100 USD to B.

Consensus and Validating Servers: Each validating server verifies the proposed changes to the last ledger; changes that are agreed by at least 50 % of the servers are packaged into a new proposal which is sent to other servers in the network. This process is re-iterated with the vote requirements increasing to 60 %, 70 %, and 80 % after which the server validates the changes and alerts the network of the closure of the last ledger. At this point, any transaction that has been performed but did not appear in the ledger is discarded and can be considered as invalid by Ripple users. Each validating server maintains a list of trusted servers known as Unique Node List (*UNL*); servers only trust the votes issued by other servers which are contained in their *UNL*. We detail and analyze Ripple's consensus protocol in Sect. 2.3.

Currently, 5 Ripple validating servers are run by Ripple Labs [7]; note however, that any entity can run its own server [34] (e.g., Snapswap [8]). By doing so, Ripple enables different institutions (e.g., banks which run their own servers) to reach a consensus with respect to the fate of financial transactions. For instance, in September 2014, Ripple Labs sealed a partnership agreement with two US banks which agreed to adopt Ripple's open-source distributed transaction infrastructure [9].

2.2 Ripple Transactions

Ripple currently supports six types of transactions [35], namely:

Payment: This is the most common type of transactions, and allows an entity to send funds from one account to another.

AccountSet: This transaction allows an entity to set options relevant for one's account. Notice that an `AccountSet` transaction enables the cancellation of a transaction with the same `SequenceNumber` provided that the transaction has not been incorporated yet in a validated ledger.

SetRegularKey: This transaction allows an entity to change/set the key used by the entity to sign future transactions.

OfferCreate: This transaction expresses an intent to exchange currencies.

OfferCancel: This transaction removes an offer from the ledger.

TrustSet: This transaction creates (or modifies) a trust link between two accounts.

As shown in Table 1, all six transaction types contain some common fields. Notice that for any entity to open an account in Ripple, it has to issue a payment with a value larger than the minimum XRP (i.e., 20 XRPs) to an account number which does not exist yet. Once this transaction is processed, a new `AccountRoot` node will be added to the global ledger to reflect the newly-created account.

2.3 The Consensus Protocol

As mentioned earlier, Ripple's consensus protocol is an asynchronous round-based protocol which is executed by the network's validating servers. At the end of every round, a new last closed ledger is published by all involved servers. The consensus protocol comprises three phases: the collection phase, the consensus phase, and the ledger closing phase.

In the collection phase, the validating servers collect the transactions that they receive from the network. Recall that transactions are typically broadcasted in the network. Upon receiving a transaction, validating servers check its authenticity; for that purpose, they verify the issuer's public key (from the ledger), and they check the validity of the corresponding signature. Transactions which come equipped with valid signatures are temporarily stored in the *candidate set CS* for subsequent validation. The validating servers then check the correctness of transactions stored in *CS*; this includes verifying that enough credit is available in the issuing account by going over the history of all transactions pertaining to that account (in case of an XRP transactions), or the existence of a trust path between the sender and receiver (in case of an IOU payment), etc. Each validating server packages validated transactions in an (authenticated) proposal and broadcasts its proposal in the network. In Ripple, this is achieved by constructing a hash tree of all validated transactions, and subsequently signing the root of the tree.

When validating server v receives a new proposal from the network, it checks that the proposal's issuer is a server which appears in its *UNL* and verifies the

Table 1. Common fields contained in all Ripple transaction types.

Field	Internal Type	Description
Account	Account	The unique address of the account that initiated the transaction
AccountTxnID	Hash256	(Optional) Hash value identifying another transaction. This field allows the chaining of two transactions together, so that a current transaction is only valid unless the previous one (by Sequence Number) is also valid and matches the hash
Fee	Amount	(Required) Integer amount of XRP, in drops, to be destroyed as a fee for distributing this transaction to the network
Flags	UInt32	(Optional) Set of bit-flags for this transaction
LastLedgerSeq	UInt32	(Optional) Highest ledger sequence number that a transaction can appear in
Memos	Array	(Optional) Additional information used to identify this transaction
Sequence	UInt32	(Required) A transaction is only valid if the sequence number is exactly 1 greater than the last-validated transaction from the same account
SigningPubKey	PubKey	(Required) ASCII representation of the public key that corresponds to the private key used to sign this transaction
SourceTag	UInt32	(Optional) Arbitrary integer used to identify the reason for this payment
TransactionType	UInt16	The type of transaction
TxnSignature	VariableLength	(Required) Transaction signature

correctness of the transactions included in the received proposal. In the positive case, these transactions are included into the locally managed transactions list TL_v. Moreover, the server maintains a vote list $Vote_t$ for every transaction t. This list is updated according to the received proposal. That is, if the transaction t is part of the proposal received from a server w ($t \in TL_v$ and $w \in UNL_v$), v will register t in $Vote_t$.

During the consensus phase, a validating server continuously processes and sends proposals. Here, the validating server only sends proposals which are agreed by more than θ percent of the servers in its *UNL*. This threshold value θ is initially set to 50 % and is gradually increased in each iteration by 10 % – until a proposal reaches consensus from 80 % of the servers in the *UNL*. Iterations are triggered by a local timer maintained by each validating server.

As shown in Algorithm 1, once a transaction t reaches 80 % acceptance, it will be removed from the candidate set, checked for double-spending (i.e., by checking

$L \leftarrow PreviousLedger$
foreach $t \in TL_v$ **do**
  if $\left(\frac{|\text{Vote}_t|}{|UNL_v|} \geq 0.8\right)$ **then**
    if $t \notin L$ **then**
      L.apply(t)
    $CS_v \leftarrow CS_v \setminus \{t\}$
  $TL_v \leftarrow TL_v \setminus \{t\}$
  $\text{Vote}_t \leftarrow \theta$
end
$\sigma_L \leftarrow Sign(H(L))$
Broadcast (L, σ_L)
foreach $u \in UNL_v$ **do**
  Receive (L_u, σ_{L_u})
end
Find the ledger L' among L_u's with valid signature which has clear majority (more than 80 %)
$CurrentLedger \leftarrow L'$

Algorithm 1. Closing the ledger

against the transactions included in the ledger). This transaction will be then appended to the ledger (L.apply(t)), and the balance of the sender/recipient will be appropriately updated. Each validating server v will forward a signed hash of its version of L in the network. A ledger is considered validated (and closed) by server v when a clear majority 80 % of validating servers which are contained in v's *UNL* also sign the same ledger L. After closing the ledger, transactions which have been received during the consensus phase will be processed, and the next round will start.

2.4 Ripple Vs. Bitcoin

In what follows, we briefly discuss the security and privacy provisions of Ripple in relation to the well-investigated Bitcoin system.

Security: Similar to Bitcoin, Ripple relies on ECDSA signatures to ensure the authenticity and non-repudiation of transactions in the system. Furthermore, since Ripple is an open payment system (like Bitcoin), all transactions and their orders of execution are publicly available. This ensures the detection of any double-spending attempt (and of malformed transactions). In Ripple, validating servers check the log of all transactions in order to select and vote for the correct transactions in the system. In this way, Ripple adopts a voting scheme across all validating servers (one vote per each validating server); the transactions for which (80 % of) the validators agree upon are considered to be valid [36]. Ripple Labs claim it is easy to identify colluding validators and recommend users to choose a set of heterogenous validators which are unlikely to be coerced as a group and are unlikely to collude.

Notice that if validators refuse to come to a consensus with each other, this is detectable by other validators, which then pronounce the network broken. In this case, the only way to resolve the problem would be to manually analyze the signed validations and proposals to see which validators were being unreasonable and for all honest participants to remove those validators from the *UNLs* (i.e., from the lists of validators they try to come to a consensus with). As far as we are aware, there is no formal security treatment of the correctness of Ripple's consensus protocol; this protocol has recently received some criticism [13,25]. In Sect. 3, we show that the current choice of parameters does not prevent the occurrence of forks in the system, and we give a necessary and sufficient condition to prevent any fork in the system.

In contrast, Bitcoin security has been thoroughly investigated in numerous studies, and as such is better understood than Ripple. In Bitcoin, transaction security is guaranteed by means of Proof of Work (PoW) which replaces the vote per validating server notion of Ripple, with a vote per computing power of the miners that are solving the PoW. Unlike Ripple, once transactions are confirmed in the global ledger (i.e., once transactions receive six confirmation blocks), it is computationally infeasible to modify these transactions [30]. In contrast, in Ripple, if at any instant in time the majority of the validating servers becomes malicious, then they can rewrite the entire history of transactions in the system. Recall that, at the time of writing, there are only a handful of Ripple validating servers which are mostly maintained by the Ripple Labs; if these servers are compromised, then the security of Ripple is at risk.

Fast Payments: In Bitcoin, payments are confirmed by means of PoW in Bitcoin blocks every 10 min on average. A study in [26] has shown that the generation of Bitcoin blocks follows a geometric distribution with parameter 0.19. This means that, since transactions are only confirmed after the generation of six consecutive blocks, then a payment is only confirmed after 1 hour on average. Although Bitcoin still recommends merchants to accept fast payments—where the time between the exchange of currency and goods is short (i.e., in the order of few seconds), several attacks have been reported against fast payments in Bitcoin [26]; a best-effort countermeasure has also been included in the Bitcoin client [26].

Unlike Bitcoin, Ripple inherently supports fast payments. As shown in Fig. 3(a), almost all ledgers are closed within few seconds; this also suggests that payments in Ripple can be verified after few seconds from being executed.

Privacy and Anonymity: Ripple and Bitcoin are instances of open payment systems. In an open payment system, *all* transactions that occur in the system are publicly announced. Here, user anonymity is ensured through the reliance on pseudonyms and/or anonymizing networks, such as TOR [15]. Users are also expected to have several accounts (corresponding to different pseudonyms) in order to prevent the leakage of their total account balance. Notice that, in Bitcoin, transactions can take different inputs, which originate from different accounts. This is not the case in Ripple, in which payments typically have a single account as input.

Although user identities are protected in Ripple and Bitcoin, the transactional behavior of users (i.e., time and amount of transactions) is leaked in the process—since transactions are publicly announced in the system. In this respect, several recent studies have shown the limits of privacy in open payment systems [11,31,37]. There are also several proposals for enhancing user privacy in these systems; most proposals leverage zero-knowledge proofs of knowledge and cryptographic accumulators in order to prevent tracking of expenditure in the network [10,28]. Although most of these studies focus on the Bitcoin system, we argue that they equally apply to Ripple. Recently, a secure privacy-preserving payment protocol for credit networks which provides transaction obliviousness has been proposed [29].

Clients, Protocol Update, and Maintenance: Both Ripple and Bitcoin are currently open source, which allows any entity to build and release its own software client to interface with either systems. The official clients for Bitcoin and Ripple are however maintained and regularly updated by the Bitcoin foundation, and Ripple Labs respectively. Bitcoin clients can also run on resource-constrained devices such as mobile phones—owing to the simple payment verification of Bitcoin [30]. As far as we are aware, there exists no secure lightweight version of Ripple.

Notice that all changes to the official Bitcoin client are publicly discussed in online forums, well justified, and voted on amongst Bitcoin developers [18]. This process is however less transparent in Ripple.

((De-)Centralized Deployment: Ripple and Bitcoin leverage completely decentralized protocols. Nevertheless, a recent study has shown the limits of decentralization in the current deployment of Bitcoin; here, it was shown that only a handful of entities can control the security of all Bitcoin transactions [18].

We argue that the current deployment of Ripple is also centralized. At the time of writing, most validating servers are run by Ripple Labs. Although there are few other servers that are run by external entities, the default list of validating servers for all clients point to the ones maintained by Ripple Labs. This also suggests that Ripple Labs can control the security of all transactions that occur in the Ripple system. Moreover, Ripple Labs and its founders retain a considerable fraction of XRPs; this represents the largest holdback of any crypto-currency [4] and suggests that Ripple Labs can currently effectively control Ripple's economy. We contrast this to Bitcoin, where the current system deployment is not entirely decentralized, yet the entities which control the security of transactions, the protocol maintenance and update, and the creation of new coins are distinct [18]. In Ripple, the same entity, *Ripple Labs*, controls the fate of the entire system.

3 Analysis of Forking in Ripple

The security of Ripple relies on the fact that the majority of the validating servers are honest and correctly verify all the received transactions. Here, ledgers fork constitute a major threat to the correct operations of the system. Forks can

occur if two conflicting ledgers get clear majority votes, and could lead to double-spending attacks [26].

Ripple claims that forks cannot occur if the *UNL* of any two servers u and v intersect in at least 20 % of the remaining validating servers ID [38]:

$$|UNL_u \cap UNL_v| \geq \frac{1}{5} \max\{|UNL_u|, |UNL_v|\} \forall u, v. \tag{1}$$

Recently, several forks [13,25] however lead to serious concerns about the correctness of the Ripple consensus protocol and the requirements for forks in the system. In what follows, we take a second look at the conditions for which a fork can occur in Ripple. More precisely, we investigate the values $w_{u,v}$, such that:

$$|UNL_u \cap UNL_v| \geq w_{u,v}(\max\{|UNL_u|, |UNL_v|\}) \forall u, v. \tag{2}$$

Notice that in the current specification of Ripple, $w_{u,v} = 0.2$ is required. We now show that this threshold is not sufficient to prevent forks in the system by means of a counter-example. Namely, consider the situation where $|UNL_u| = |UNL_v| = 5$ and $|UNL_u \cap UNL_v| = 2$. Obviously, it holds that $|UNL_u \cap UNL_v| = 0.4 \cdot \max\{|UNL_u|, |UNL_v|\}$. Assume now that one server in $UNL_u \cap UNL_v$ votes for L_1 and the other for (conflicting ledger) L_2. Moreover, assume that all servers in $UNL_u \setminus UNL_v$ vote for L_1 and similarly all servers in $UNL_v \setminus UNL_u$ vote for L_2. This means that a majority of 80 % in UNL_u vote for L_1 and likewise a majority of 80 % in UNL_v vote for L_2. This clearly results in a fork in the system.

As this example shows, the condition displayed in Eq. 2 cannot prevent forks in general for values $w_{u,v} \leq 0.4$. In the following, we will prove that if the intersection set size between the *UNL* of any two servers is more than 40 % of size of the largest *UNL*, that is $w_{u,v} > 0.4$, then forks in Ripple are impossible. The consequence is that forks in Ripple are impossible *if and only if*

$$|UNL_u \cap UNL_v| > 0.4 \cdot \max\{|UNL_u|, |UNL_v|\} \forall u, v. \tag{3}$$

For the sake of readability, we denote the threshold value for any transaction to get clear majority votes by ρ where $0.5 < \rho \leq 1$. We then prove that forks are not possible if $w_{u,v} > \rho/2$ for any servers u and v.

Recall that a fork refers to the situation that two different validating servers u and v agree on conflicting ledgers $L_1 \neq L_2$. This means that at least a fraction ρ of servers in UNL_u agree on ledger L_1 and at least a fraction ρ of servers in UNL_v agree on ledger L_2. We consider the following sets:

$$A := UNL_u \setminus UNL_v, \quad B := UNL_u \cap UNL_v, \quad C := UNL_v \setminus UNL_u. \tag{4}$$

For each server contained in $UNL_u \cup UNL_v$, three possible cases my occur:

Case 1: The server publishes ledger L_1.
Case 2: The server publishes ledger L_2.
Case 3: The server does not reply or publishes any other ledger besides L_1 and L_2.

In the sequel, we denote by A_1 the subset of servers in set A publishing L_1, by A_2 the subset of servers in A publishing L_2, and by A_3 the subset of servers publishing neither L_1 nor L_2. Clearly, A_1, A_2 and A_3 are mutually exclusive, and $|A_1| + |A_2| + |A_3| = |A|$. Analogously, we define sets B_1, B_2, B_3, C_1, C_2, and C_3 (cf. Eq. 4).

Necessary Conditions for Forking: According to the specification of Ripple, it holds that if more than a fraction ρ of the servers present in any server's *UNL* publishes the same validation ledger hash, that ledger will be accepted by that server. Hence,

1. Ledger L_1 will be accepted by server u if and only if

$$\begin{aligned} |A_1| + |B_1| &\geq \rho(|A_1| + |A_2| + |A_3| + |B_1| + |B_2| + |B_3|) \\ \Leftrightarrow (1-\rho)(|A_1| + |B_1|) &\geq \rho(|A_2| + |A_3| + |B_2| + |B_3|) \\ \Leftrightarrow |A_1| + |B_1| &\geq \frac{\rho}{1-\rho}(|A_2| + |A_3| + |B_2| + |B_3|) \end{aligned} \tag{5}$$

2. Likewise, ledger L_2 will be accepted by server v if and only if

$$|B_2| + |C_2| \geq \frac{\rho}{1-\rho}(|B_1| + |B_3| + |C_1| + |C_3|) \tag{6}$$

Minimum Intersection Size: Notice that a fork is only possible if both Eqs. 5 and 6 are satisfied. Assuming that $|UNL_u \cap UNL_v| > w_{u,v} \max\{|UNL_u|, |UNL_v|\} \forall u, v$, we show in what follows that $w_{u,v} \geq 0.4$ ensures that no fork can occur in Ripple.

Observe that:

$$\begin{aligned} |UNL_u \cap UNL_v| &> w_{u,v} \cdot |UNL_u| \\ |B_1| + |B_2| + |B_3| &> w_{u,v}(|A_1| + |A_2| + |A_3| + |B_1| + |B_2| + |B_3|) \\ (1 - w_{u,v})(|B_1| + |B_2| + |B_3|) &> w_{u,v}(|A_1| + |A_2| + |A_3|) \\ (|B_1| + |B_2| + |B_3|) &> \frac{w_{u,v}}{1 - w_{u,v}}(|A_1| + |A_2| + |A_3|) \end{aligned} \tag{7}$$

Similarly, we have:

$$(|B_1| + |B_2| + |B_3|) > \frac{w_{u,v}}{1 - w_{u,v}}(|C_1| + |C_2| + |C_3|) \tag{8}$$

Now, adding Eqs. (7) and (8) we get,

$$(|B_1| + |B_2| + |B_3|) > \frac{w_{u,v}}{2(1 - w_{u,v})}(|A_1| + |A_2| + |A_3| + |C_1| + |C_2| + |C_3|) \tag{9}$$

Assuming that both Eqs. 5 and 6 are satisfied, it follows that:

$$\begin{aligned} |A_1| + |B_1| + |B_2| + |C_2| &\geq \frac{\rho}{1-\rho}(|A_2| + |B_2| + |B_1| + |C_1| + |A_3| + |C_3|) \\ &\quad + \frac{2\rho}{1-\rho}|B_3| \\ |A_1| + |C_2| &\geq \frac{\rho}{1-\rho}(|A_2| + |C_1| + |A_3| + |C_3|) \\ &\quad + \frac{2\rho - 1}{1-\rho}(|B_1| + |B_2| + |B_3|) + \frac{1}{1-\rho}|B_3|. \end{aligned} \tag{10}$$

Combining Eqs. 9 and 10, we get the following strict inequality:

$$
\begin{aligned}
|A_1| + |C_2| > & \frac{\rho}{1-\rho}(|A_2| + |C_1| + |A_3| + |C_3|) \\
& + \frac{(2\rho-1)w_{u,v}}{2(1-\rho)(1-w_{u,v})}(|A_1| + |A_2| + |A_3| + |C_1| + |C_2| + |C_3|) \\
& + \frac{1}{1-\rho}|B_3|
\end{aligned}
$$

This can be rephrased to:

$$
\begin{aligned}
& (1 - \frac{(2\rho-1)w_{u,v}}{2(1-\rho)(1-w_{u,v})}) > \\
& \underbrace{\frac{1}{(|A_1| + |C_2|)}}_{\geq 0} \cdot \left[(\underbrace{\frac{\rho}{1-\rho}}_{\geq 0} + \underbrace{\frac{(2\rho-1)w_{u,v}}{2(1-\rho)(1-w_{u,v})}}_{\geq 0}) \underbrace{(|A_2| + |A_3| + |C_1| + |C_3|)}_{\geq 0} + \underbrace{\frac{1}{1-\rho}|B_3|}_{\geq 0} \right]
\end{aligned}
$$

As already marked, the right-hand side is ≥ 0. Hence, this cannot hold if:

$$
\begin{aligned}
(1 - \frac{(2\rho-1)w_{u,v}}{2(1-\rho)(1-w_{u,v})}) &\leq 0 \\
(2-2\rho)(1-w_{u,v}) - (2\rho-1)w_{u,v} &\leq 0 \\
(2-2\rho-w_{u,v}) &\leq 0 \\
w_{u,v} &\geq 2(1-\rho)
\end{aligned}
$$

In consequence, if $|UNL_i \cap UNL_j| > 2(1-\rho)\max\{|UNL_i|, |UNL_j|\} \forall i, j$, then no fork can occur in Ripple for sure. Since $\rho = 0.8$ in the current Ripple system, a sufficient condition for preventing forks is to ensure $w_{u,v} > 0.4$ for all servers u and v.

4 Ripple Under the Hood

In this section, we study the current deployment of Ripple. For that purpose, we extract relevant statistics about the use of Ripple in the period from January 2013 till January 2015.

At the time of writing, there are more than 12 million ledgers starting from January 2013 [33]. Ripple also claims to have little above 150,000 accounts with an average of almost 170 accounts created per day since the launch of the system.

To better understand the current usage and dynamics in Ripple, we built a parser using Java, which uses the Websocket protocol to download and parse ledgers created in the period between January 2013 and January 2015 from the main Ripple server[1], and from three auxiliary servers[2]. Our parser leverages the

[1] Available from s1.ripple.com.

[2] Available from s-east.ripple.com, and s-west.ripple.com.

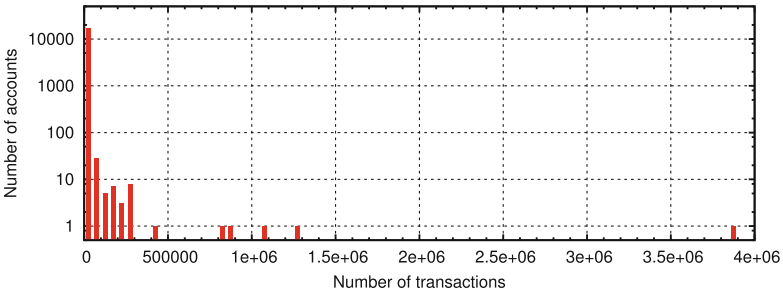


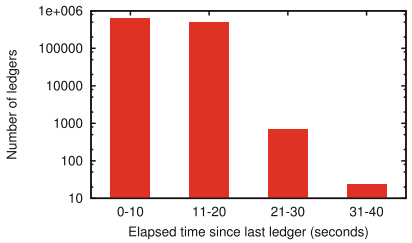
Fig. 2. Distribution of the number of transactions per address in Ripple in January/February 2015.

Tyrus library [3] and a connection pool to access a local MySQL database which stores information acquired from the downloaded ledgers. For ease of presentation, we divide the period of study into 5 different time intervals comprising of 2 months each. In total, we parsed a total of 4,645,799 ledgers comprising over 33,304,766 transactions, and 153,637 total accounts.

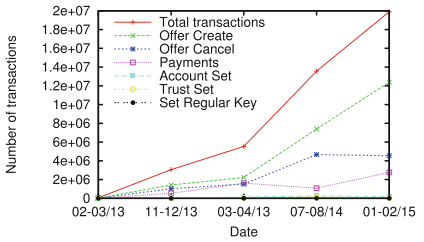
Transactions per account: Figure 2 depicts the distribution of transactions per Ripple account in the parsed ledgers. Our results show that most ($> 99\,\%$) Ripple accounts have performed very few transactions in the system. Notice that this does not necessarily provide evidence that Ripple users are inactive; for example, privacy-aware users could set up, in theory, different accounts for each transaction they perform in order to prevent the leakage of their total balance in the system [11].

Ledger closing time: In Fig. 3(a), we measure the time elapsed between the creation of two successive ledgers in the time interval spanning across January and February 2015. Our results show that indeed most ledgers are finalized in few seconds; while we observe that some ledgers take around 30–40 s to close, almost 99 % of the ledgers created in the first two months of 2015 were closed in less than 20 s.

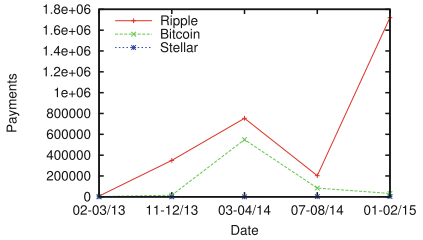
Transactions dynamics over time: In Fig. 3(b), we compute the number of performed Ripple transactions over time. Our findings show that the number of transactions performed in the Ripple system has been steadily increasing over time. For instance, in the first two months of 2015, more than 20,000,000 Ripple transactions have been executed. Our findings however indicate that more than 60 % of these transactions correspond to Offers in the system—and not to actual payments—while OfferCancel transactions correspond to 20 % of the total transactions in the system. Payment transactions comprise less than 15 % of the total transactions in the system, and are only increasing marginally over time. For example, there are almost 33,000 payment transactions per day, on average, starting from March 2014 and until February 2015. Our results also show that there were a total of 6765 distinct accounts whose trust had been extended to using TrustSet transactions.



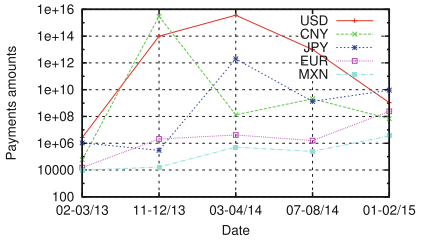
(a) Closure times of ledgers in January/February 2015.



(b) Evolution of the number of Ripple transactions over time.



(c) Evolution of the number of XRP payments and trade of digital currencies over time.



(d) Evolution of the trade of fiat currencies over time.

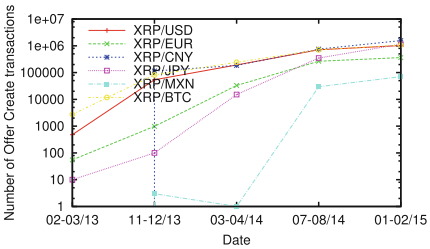
Fig. 3. Characterization of the Ripple system in the period between January 2013 and January 2015.

In Fig. 3(c) and (d), we further analyze the payment transactions performed in the Ripple system; our findings show that direct XRP to XRP transactions comprise the majority of transactions performed in Ripple. For example, in the first two months of 2015, there were almost 2 million payments in Ripple (cf. Fig. 3(b)); as shown in Fig. 3(c), almost 1.8 million of those correspond to direct XRP transactions.

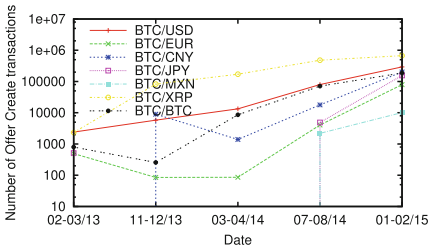
Although Ripple was used as a medium to exchange BTCs in March/April 2014, we further remark that Bitcoin trade in Ripple has considerably shrunk in the first two months of 2015 to less than 1 % of the performed payments. Moreover, in July/August 2014, our findings suggest that the Ripple system has witnessed a considerable setback in the number of direct XRP transactions, and in the trade of digital currencies, such as Bitcoin. We also remark that other digital currencies, such as Stellar, are rarely traded in the Ripple system.

In terms of the trade of fiat currencies, our results show that trading of fiat currencies represents almost 10 % of the actual payments in Ripple in the start of 2015. However, as shown in Fig. 3(d), our findings suggest that extremely large amounts of fiat currencies are being traded in Ripple. For instance, we measure the trading of almost $1 \cdot 10^{16}$ USD in March/April 2014. Our results show that only a handful of payments trade such obscene amounts; we believe that these payments are not actual payments, but could result from testing/debugging in the system[3].

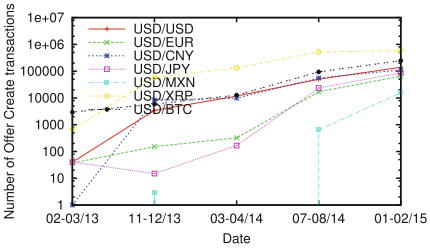
[3] Recall that Ripple has no means to enforce the execution of payments.



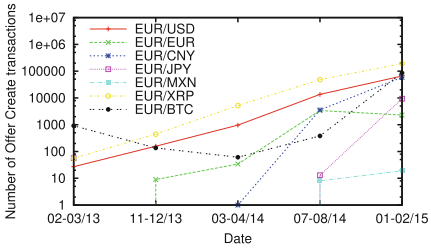
(a) Evolution of XRP-based trade over time.



(b) Evolution of BTC-based trade over time.



(c) Evolution of USD-based trade over time.



(d) Evolution of EUR-based trade over time.

Fig. 4. Characterization of IOU payments in the Ripple system over time starting from February 2013.

OfferCreate evolution: Figure 4(a), (b), (c), and (d) depict the distribution of OfferCreate transactions in the system. Recall that these transactions comprise almost 60 % of Ripple transactions, and are mainly performed by the market makers that populate the system. Our findings suggest that, as expected, the biggest market makers offer the trading of XRP to BTCs, USD, and EUR. Additional market makers offering the trade of XRP to CNY and JPY emerged starting from November 2013, and March 2014, respectively. There are also a considerable number of Offers for trading major fiat currencies such as USD and EUR. Although the total number of offers is growing over time, we do not find evidence for growth of the corresponding Ripple payments.

Summary of findings: In summary, our results suggest that—although it has been introduced almost 2 years ago—Ripple is still far from being used as a trade platform. Ripple advertises a large number of active accounts [33]. However, we do not find strong evidence that users are active in Ripple; most accounts contain a small number of XRPs—which users e.g., could have received from the one of the many giveaways organized by Ripple Labs [32]. Moreover, although the number of transactions in Ripple seems to be considerably increasing over time, most of the transactions in the system (>70 %) correspond to OfferCreate and OfferCancel transaction types. The number of actual payments in the system is only marginally increasing over time, and is dominated by direct XRP payments. Finally, although there are a number of currency exchanges performed

via Ripple—some of which deal with huge amounts—it is hard to tell whether those transactions have been actually concluded since the Ripple system has no way to enforce IOU transactions.

5 Related Work

Although Bitcoin and its many variants have received considerable attention in the literature, there are surprisingly no studies—as far as we are aware—which analyze Ripple.

Bonneau *et al.* [12] provide a comprehensive exposition of the second generation crypto-currencies, including Bitcoin and the many alternatives that have been implemented as alternate protocols. However, this work does not provide any insights on the Ripple protocol.

In [26,27], Karame *et al.* thoroughly investigate double-spending attacks in Bitcoin and show that double-spending fast payments in Bitcoin can be performed in spite of the measures recommended by Bitcoin developers. In [11,17], the authors evaluate user privacy in Bitcoin and show that Bitcoin leaks considerable information about users. In [31], Ober *et al.* studied the time-evolution properties of Bitcoin. In [29], Moreno-Sanchez *et al.* propose a provably secure privacy-preserving payment protocol for credit networks, such as Ripple.

6 Conclusion

In this paper, we studied the current deployment of the Ripple payment system. We showed that although Ripple leverages a decentralized consensus protocol, the current deployment of Ripple is not decentralized, and offers unconditional power for Ripple Labs to control the fate and security of all Ripple transactions.

We also showed that the currently adopted assumptions to prevent the occurrence of forks in the system are insufficient. Namely, our findings show that the intersection set size between the *UNL* of any two validating servers needs to be more than 40 % of the maximum *UNL* set size in order to ensure the absence of any fork in the system. Finally, we analyzed the current usage of the Ripple system; our results show that most users in Ripple seem inactive, and that Ripple is still not being widely used as a trade platform.

Our results motivate the need for a rigorous analysis of the Ripple system prior to any large scale deployment. We therefore hope that our findings solicit further research in this area.

Acknowledgements. The authors would like to thank Ludovic Barman for the help in extracting the relevant statistics from the Ripple ledgers.

References

1. Litecoin: Open source P2P internet currency. https://litecoin.org/
2. Namecoin: A trust anchor for the internet. https://namecoin.info/
3. Project tyrus. https://tyrus.java.net/
4. Ripple. http://en.wikipedia.org/wiki/Ripple_%28payment_protocol%29
5. Ripple labs circling 30m$ in funding. http://www.pymnts.com/news/2015/ripple-labs-circling-30m-in-funding/#.VRLnJfnF98F
6. Ripple: Opening access to finance. https://ripple.com/
7. Ripple validating servers. https://ripple.com/ripple.txt
8. Snapswap Ripple gateway. https://snapswap.us/#/
9. US banks announce Ripple protocol integration. http://www.coindesk.com/us-banks-announce-ripple-protocol-integration/
10. Androulaki, E., Karame, G.O.: Hiding transaction amounts and balances in Bitcoin. In: Trust and Trustworthy Computing - 7th International Conference, TRUST 2014, Heraklion, Crete, Greece, 30 June – 2 July, 2014. Proceedings, pp. 161–178 (2014)
11. Androulaki, E., Karame, G.O., Roeschlin, M., Scherer, T., Capkun, S.: Evaluating user privacy in bitcoin. In: Sadeghi, A.-R. (ed.) FC 2013. LNCS, vol. 7859, pp. 34–51. Springer, Heidelberg (2013)
12. Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J.A., Felten, E.W.: Research perspectives and challenges for Bitcoin and cryptocurrencies. In: 2015 IEEE Symposium on Security and Privacy, May 2015
13. Buterin, V.: Bitcoin network shaken by blockchain fork. https://bitcoinmagazine.com/3668/bitcoin-network-shaken-by-blockchain-fork/
14. Coinist Inc., Ripple gateways. https://coinist.co/ripple/gateways
15. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. In: Proceedings of the 13th Conference on USENIX Security Symposium, SSYM 2004, Berkeley, CA, USA, vol. 13, p. 21. USENIX Association (2004)
16. Elias, M.: Bitcoin: Tempering the digital ring of gyges or implausible pecuniary privacy (2011). http://ssrn.com/abstract=1937769
17. Gervais, A., Capkun, S., Karame, G.O., Gruber, D.: On the privacy provisions of bloom filters in lightweight bitcoin clients. In: Proceedings of the 30th Annual Computer Security Applications Conference, ACSAC 2014, 8–12 December, 2014, New Orleans, LA, USA, pp. 326–335 (2014)
18. Gervais, A., Karame, G.O., Capkun, V., Capkun, S.: Is Bitcoin a decentralized currency? IEEE Secur. Priv. **12**(3), 54–60 (2014)
19. Ghosh, A., Mahdian, M., Reeves, D.M., Pennock, D.M., Fugger, R.: Mechanism design on trust networks. In: Deng, X., Graham, F.C. (eds.) WINE 2007. LNCS, vol. 4858, pp. 257–268. Springer, Heidelberg (2007)
20. International Ripple Business Association. Listed businesses. http://www.xrpga.org/listed-businesses.html
21. International Ripple Business Association. Ripple exchangers. http://www.xrpga.org/exchangers.html
22. International Ripple Business Association. Ripple gateways. http://www.xrpga.org/gateways.html
23. International Ripple Business Association. Ripple market makers. http://www.xrpga.org/market-makers.html
24. International Ripple Business Association. Ripple merchants. http://www.xrpga.org/merchants.html

25. Joyes, K.: Safety, liveness and fault tolerance - the consensus choices. https://www.stellar.org/blog/safety_liveness_and_fault_tolerance_consensus_choice/
26. Karame, G.O., Androulaki, E., Capkun, S.: Double-spending fast payments in Bitcoin. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS 2012, New York, NY, USA, pp. 906–917. ACM (2012)
27. Karame, G.O., Androulaki, E., Roeschlin, M., Gervais, A., Čapkun, S.: Misbehavior in Bitcoin: a study of double-spending and accountability. ACM Trans. Inf. Syst. Secur., 18(1), 2:1–2:32 (2015)
28. Miers, I., Garman, C., Green, M., Rubin, A.D.: Zerocoin: anonymous distributed e-cash from Bitcoin. In: Proceedings of the 2013 IEEE Symposium on Security and Privacy, SP 2013, Washington, DC, USA, pp. 397–411. IEEE Computer Society (2013)
29. Moreno-Sanchez, P., Kate, A., Maffei, M., Pecina, K.: Privacy preserving payments in credit networks: Enabling trust with privacy in online marketplaces. In: Network and Distributed System Security (NDSS) Symposium (2015)
30. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2009). http://bitcoin.org/bitcoin.pdf
31. Ober, M., Katzenbeisser, S., Hamacher, K.: Structure and anonymity of the Bitcoin transaction graph. Future Internet **5**(2), 237–250 (2013)
32. Ripple Labs Inc., Giveaways - XRPtalk. https://xrptalk.org/forum/105-giveaways/
33. Ripple Labs Inc., Ripple charts. https://www.ripplecharts.com
34. Ripple Labs Inc., Setup a validating server. https://wiki.ripple.com/Setup_a_validating_server
35. Ripple Labs Inc., Transactions. https://ripple.com/build/transactions/
36. Ripple Labs Inc., Why is Ripple not vulnerable to Bitcoin's 51 % attack?
37. Ron, D., Shamir, A.: Quantitative analysis of the full bitcoin transaction graph. In: Sadeghi, A.-R. (ed.) FC 2013. LNCS, vol. 7859, pp. 6–24. Springer, Heidelberg (2013)
38. Schwartz, D., Youngs, N., Britto, A.: The Ripple protocol consensus algorithm (2014). https://ripple.com/files/ripple_consensus_whitepaper.pdf