# GOLDRIUM

## The Only Cryptocurrency that is PCI compliant

WHITE PAPER

## PLEASE READ THIS SECTION CAREFULLY. IF YOU ARE IN ANY DOUBT AS TO THE ACTION YOU SHOULD TAKE, YOU ARE ADVISED TO CONSULT YOUR LEGAL, FINANCIAL, TAX, OR OTHER PROFESSIONAL ADVISOR(S).

The information listed below may not be exhaustive and does not imply any element of the contractual relationship. While we strive to ensure that all elements of this document are accurate and up-to-date, this material does not constitute professional advice. Goldrium does not warrant or accept any legal liability arising from the accuracy, reliability, timeliness or completeness of any content in this document. Contributors, sponsors and potential Goldrium holders must seek appropriate independent professional advice before relying on any material-based obligation or transaction published in this document for which material is published for reference only. Goldrium investments are not intended to create titles in any jurisdiction. This document does not constitute a prospectus or an offer of any kind and does not imply that it is an offer of securities or an application for investment in securities in any jurisdiction whatsoever.

## IMPORTANT NOTICE:

Read the following note carefully before reading this document prepared by the company ("Whitepaper"). This notice applies to everyone who has read this document. Please note that this notice can be modified or updated. The white paper of Goldrium. No shares or other securities of the Company are offered for subscription or sale in any jurisdiction in accordance with the Whitepaper. The whitepaper is publicly available for information purposes only and does not require any action by the public or shareholders of the Company. The Whitepaper does not constitute an offer or invitation to anyone to enter or register shares or other securities in the Company. Currently, the

Company's shares are not offered for registration under the Securities Act of any country or a securities law of any country. No one is obligated to enter into any contract or legal obligation to sell, purchase, sponsor or contribute to the growth of Goldrium.

# FORWARD LOOKING STATEMENTS

Some of the statements in the Whitepaper include forward-looking statements which reflect Goldrium's current views with respect to product development, execution roadmap, financial performance, business strategy and future plans, both with respect to the Company and the sectors and industries in which the Company operates. Statements which include the words "expects", "intends", "plans", "believes", "projects", "anticipates", "will", "targets", "aims", "may", "would", "could", "continue" (and any conjugation thereof) and any similar statements are of a future or forward-looking nature. All forward-looking statements address matters that involve risks and uncertainties. Accordingly, there are or will be important factors that could cause the actual results to differ materially from those indicated in these statements. These factors include but are not limited to those described in the part of the Whitepaper entitled "Risk Factors", which should be read in conjunction with the other cautionary statements that are included in the Whitepaper. Any forward-looking statements in the Whitepaper reflect the current views with respect to future events and are subject to these and other risks, uncertainties and assumptions relating to the operations, results of operations and growth strategy. These forward-looking statements speak only as of the date of the Whitepaper. Subject to industry acceptable disclosure and transparency rules and common practices, the Company

undertakes no obligation publicly to update or review any forward-looking statement, whether as a result of new information, future developments or otherwise. All subsequent written and oral forward-looking statements attributable to the Company or individuals acting on behalf of the Company are expressly qualified in their entirety by this paragraph. Prospective buyers or Crowdfunding contributors or sponsors of Goldrium should specifically consider the factors identified in the Whitepaper, which could cause actual results to differ before making a purchase, contribution or sponsorship decision. No statement in the Whitepaper is intended as a profit forecast and no statement in the Whitepaper should be interpreted to mean that the earnings of the Company for the current or future years would be as may be implied in this Whitepaper. I hereby acknowledge that I have read and understand the notices and disclaimers set out above.

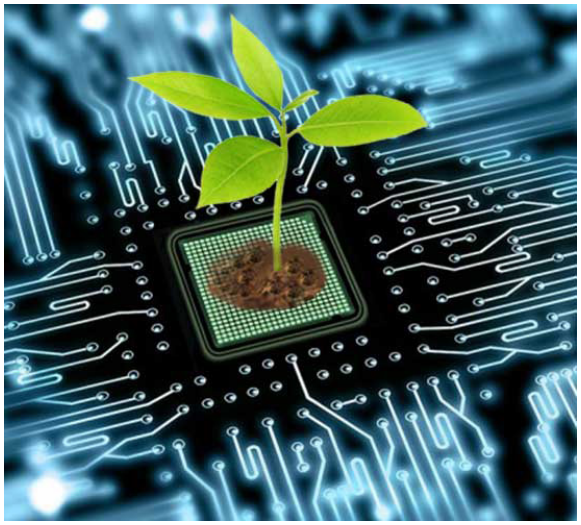
# Table Of Contents

# INTRODUCTION

Goldrium is an innovative cryptocurrency operating on its own technology with the vision to reduce the costs of the transaction. It is designed to process payments and is the only PCI compliant cryptocurrency. One of the advantages of Goldrium is that it does not rely on blockchain technology. Because it is also capable of offering tokens to be created as a new ICO. With a transaction time less than 500 ms and 100,000 transaction processing capacity simultaneously, making it an instant transaction time, this makes Goldrium the fastest cryptocurrency in the world with the goal to meet PCI standards while surpassing every kind of transaction speed.

## Etymology

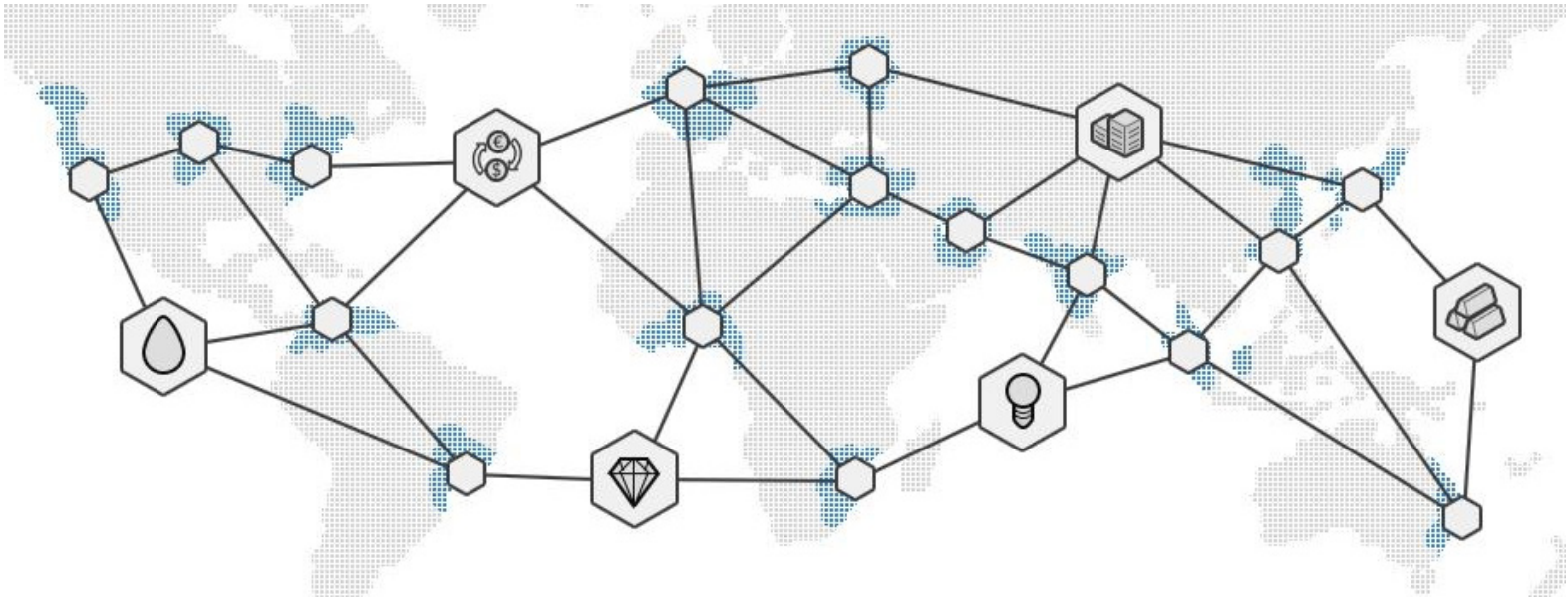
Gold was once used as currency and later it was used to back up fiat currencies. It always been "the gold standard" of excellence due to its properties and rareness. Arium or Rium is a Latin word which means a place or function of something. We combined the two words together to signify a true currency which holds gold a value in the world and has its unique place and function as a cryptocurrency.



2700+ years old

## Goldrium include the following:

- Goldrium is PCI compliant.
- Goldrium enables very fast microtransactions and trading.
- Goldrium relies on its own technology.
- It has a single-minded community and development team behind it.
- Goldrium is capable of offering tokens to be created as a new ICO.



## Goldrium's Promotional Strategy

We promote Goldrium through different methods for maximum exposure, such promotional methods include:

a) INFLUENCERS: While marketing Goldrium, we also extend beyond the usual field of influence, these foreign connections are exactly what Goldrium needs. Developing honest, authentic, and mutually beneficial relationships with influencers on different social media platforms has helped in making Goldrium more popular among people of different class and age.

b) WEBSITES: As an enterprise brand, Goldrium is prominently featured across different top-ranking sites and blogs. This helps to expose Goldrium to an even greater audience

c) SOCIAL MEDIA: Social media is pretty much one of our major method of promotion. There are two ways we connect with our audience via social- paid and organic. Below is a quick breakdown of both :


i) ORGANIC - Organic promotion on social media comes in the simple form of having a presence; e.g. a Facebook page, Twitter and Instagram profiles. We connect with our audience in a meaningful way.

ii) PAID - According to a 2016 report by Kenshoo, mobile app ad click-through rates (CTRs) went up 32% YoY, while the inverse happened to CPC (cost per click), which decreased by 33%. Investing money on Goldrium's promotion boost our discoverability and gets us more people flocking to Goldrium. Because of our broad audiences and ability to segment, Facebook and Instagram are very effective advertisement platforms for Goldrium.

d) Word of mouth: Word of mouth is still worth its weight in gold. Due to the value that Goldrium provides, Members cannot help but to invite both friends and families alike.

## Executive Summary

Goldrium is wonderful in many ways. It captures individuals with its own hybrid technology and has everyone longing to create their own token for new ICO. The Goldrium protocol will be key to promoting liquidity of crypto-assets in the cryptocurrency ecosystem and improving the public perception of tokens as a tradable asset class. The vision of the team behind Goldrium is solid, with the main focus on trustless solutions, instant transaction, and high liquidity.

The Goldrium ledger operates mainly on the SCRYPT algorithm because the Teams goal is to have this currency accepted all over the world by partnering with credit card processors, payment gateways, and much more. Goldrium was conceived because there are so many currencies in the market. The team believes that a coin should have functionalities which can cater to many communities so that an investor in a particular currency can benefit on many levels, and not

just focusing on one specific scenario.

## What Goldrium is Not

Goldrium is not a get-rich-quick scheme. Creating new ICO using the Goldrium technology will not make you an instant millionaire. Investors are focused on the long haul and use of Goldrium in making extremely fast and secure transactions.


ii) Goldrium helps to solve the following problems:

A) New ICO can be created on Goldrium technology

B) Goldrium reduces High settlement risks

C) Goldrium helps with International transactions.

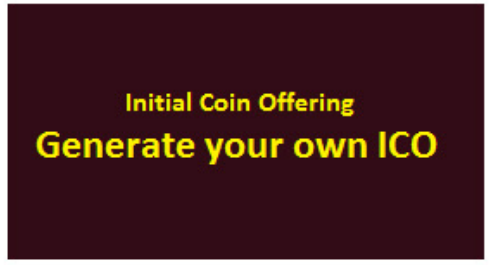
D) Goldrium helps to facilitate extremely complex supply chains

## Why is Goldrium a better payment option?

Goldrium is disruptive and is growing at an exponential rate. Goldrium as a new, easy to access cryptocurrency, will appeal toall sorts of people with various paths of life in larger numbers than we've seen with Fiat. As such, Goldrium will have great appeal for:

- Long-term holders. Those who wish to invest and hold Goldrium for any potential future value.
- Cryptocurrency toall enthusiasts toall wishing toall to be involved toall in the development toall of the toall next toall evolutionary toall step toall of toall the toall digital toall coin toall market toall .
- Users can cash out their Goldrium or do a trade in with other currencies.
- Goldrium can be used as another source of income.
- Create a new ICO with Goldrium ecosystem





## A simple explanation of cryptocurrencies

A simple explanation of cryptocurrencies Imagine a world without money. Not easy to do. We've lived with money in some form or other since civilization began. Money is a way of storing your hard work in a convenient way to exchange for someone else's hard work. We're so used to money in its modern-day form that we don't spend much time thinking about it as a concept. Modern fiat currencies such as the US Dollar and the Great British Pound are based on trust. These traditional currencies are no longer backed by anything (the gold standard ended in the US in 1971 and the UK in 1931). Trust works well in very large, stable, economies, however most of the world is made up of poor countries with less stable economies. When a government is short of money, it is awfully tempting to print some more. The problem with printing more money is that the store of value is decreased with the increase of supply. If you don't already know about hyperinflation then Google the "Weimar Republic Hyperinflation" or "Zimbabwe hyperinflation" to see what happens to a currency when too much is printed.

## Electronic Currency

Traditional currencies are made of paper and metal, however in most developed nations people can exchange these currencies between themselves electronically by relying on a third party (a bank) to store ledgers of the money. The bank stores a digital number (the amount of currency one person has) and they inform another bank that some of that money has been sent to someone else. One ledger is decreased and another increased. All of the existing digital exchange of money is done by third party trust in banks. The concept of a cryptocurrency is not just a way to transfer money between people, it is an entirely new way of thinking about money. The reason we currently need banks to make digital transfers is because of something called the Double Spending Problem. All digital things can be copied. You've heard of the film industry suffering from pirated movies. The film industry has spent years and many millions of dollars trying to prevent it, but they have failed. Digital things can be copied. This means that in the past, any digital currency suffered from the ability of users to "print more of it". The double spending problem is solved by cryptocurrency, and in solving it, has opened up a radical new way of thinking about store of value, trust and convenience. Before you dismiss this and think "banks do a perfectly good job, we don't need this" remember that.

## A simple explanation of cryptocurrencies

Why not use a bank?

Being able to transfer money to someone else without having to use a bank dramatically reduces the complexity and increases the speed of international transfers. This makes international trade easier and cheaper. With the amount of fraud that is reported online people are fearful of entering their financial information to make purchases, especially for small, low cost items. Cryptocurrencies can be anonymous and protect the payor and payee. There are over a billion people in the world that do not have access to a bank, and yet they have access to the internet. There are over 2 billion people in the world without a bank account, and mobile devices and internet access is growing fast.
a radical new way of thinking about store of value, trust and convenience. Before you dismiss this and think "banks do a perfectly good job, we don't need this" remember that.

## Why use a Goldrium?

1) Store value securely, privately and digitally.

2) Move that money around the world almost instantly with zero cost.

3) Goldrium offer a high level of security and anonymity without requiring a bank to be involved

4) Goldrium is digital and cannot be counterfeited or reversed arbitrarily by the sender, as with credit card charge-backs.

5) Everyone can access Goldrium, there are approximately 2.2 billion individuals with access to the Internet or mobile phones who don't currently have access to traditional exchange systems.

6) Goldrium solves the problem of slow transaction, while still maintaining a coin that will be increasing in value.
7) A PCI compliant cryptocurrency which is the gold standard of financial and payment industry

## Abstract

While several Scrypt algorithms exist for the Byzantine Generals Problem, specifically as it pertains to distributed payment systems, many suffer from high latency induced by the requirement that all nodes within the network communicate synchronously. In this work, we present a novel GLD algorithm that circumvents this requirement by utilizing collectivelytrusted subnetworks within the larger network. We show that the "trust" required of these subnetworks is, in fact, minimal and can be further reduced with principled choice of the member nodes. In addition, we show that minimal connectivity is required to maintain agreement throughout the whole network. The result is a low-latency GLD algorithm which still maintains robustness in the face of Byzantine failures. We present this algorithm in its embodiment in the Goldrium Protocol.

## Introduction

Interest and research in distributed GLD systems has increased markedly in recent years, with a central focus being on distributed payment networks. Such networks allow for fast, lowcost transactions which are not controlled by a centralized source. While the economic.

benefits and drawbacks of such a system are worthy of much research in and of themselves, this work focuses on some of the technical challenges.

that all distributed payment systems must face. While these problems are varied, we group them into three main categories: correctness, agreement, and utility. By correctness, we mean that it is necessary for a distributed system to be able to discern the difference between a correct and fraudulent transaction. In traditional fiduciary settings, this is done through trust between institutions and cryptographic signatures that guarantee a transaction is indeed coming from the institution that it claims to be coming from. In distributed systems, however, there is no such trust, as the identity of any and all members in the network may not even be known. Therefore, alternative methods for correctness must be utilized. Agreement refers to the problem of maintaining a single global truth in the face of a decentralized.

Accounting system. While similar to the correctness problem, the difference lies in the fact that while a malicious user of the network may be unable to create a fraudulent transaction (defying correctness), it may be able to create multiple correct transactions that are somehow unaware of each other,

and thus combine to create a fraudulent act. For example, a malicious user may make two simultaneous purchases.

with only enough funds in their account to cover each purchase individually, but not both together. Thus each transaction by itself is correct, but if executed simultaneously in such a way that the distributed network as a whole is unaware of both, a clear problem arises, commonly referred to as the "DoubleSpend Problem." Thus the agreement problem can be summarized as the requirement that only one set of globally recognized transactions exist in the network.

Utility is a slightly more abstract problem, which we define generally as the "usefulness" of a distributed payment system, but which in practice most often simplifies to the latency of the system. A distributed system that is both correct and in agreement but which requires one year to process a transaction, for example, is obviously an inviable payment system. Additional aspects of utility may include

the level of computing power required to participate in the correctness and agreement processes or the technical proficiency required of an end user to avoid being defrauded in the network. Many of these issues have been explored long before the advent of modern distributed computer systems, via a problem known as the "Byzantine Generals Problem." In this problem, a group of generals each control a portion of an army and must coordinate an attack by sending messengers to each other. Because the generals are in an unfamiliar and hostile territory, messengers may fail to reach their destination (just as nodes in a distributed network may fail, or send corrupted data instead of the intended message). An additional aspect of the problem is that some of the generals may be traitors, either individually, or conspiring together, and so messages may

arrive which are intended to create a false plan that is doomed to failure for the loyal generals (just as malicious members of a distributed system may attempt to convince the system to accept fraudulent transactions, or multiple versions of the same truthful transaction that would result in a double-spend). Thus a distributed payment system must be robust both in the face of standard failures and so-called "Byzantine" failures, which may be coordinated and originate from multiple sources in the network. In this work, we analyze one particular implementation of a distributed payment system: the Goldrium Protocol. We focus on the algorithms utilized to achieve the above goals of correctness, agreement, and utility, and show that all are met (within necessary and predetermined tolerance thresholds, which are well-understood). In addition, we provide code that simulates the GLD process with parameterizable network size, number of malicious users, and message sending latencies.

## Definitions, Formalization and Previous Work

We begin by defining the components of the Goldrium Protocol. In order to prove correctness, agreement, and utility properties, we first formalize those properties into axioms. These properties, when grouped together, form the notion of GLD: the state in which nodes in the network reach correct agreement. We then highlight some previous results relating to GLD algorithms, and finally state the goals of GLD for the Goldrium Protocol within our formalization framework.

## Goldrium Protocol Components

We begin our description of the Goldrium network by defining the following terms:

**Server:**

A server is any entity running the Goldrium Server software (as opposed to the Goldrium Client software which only lets a user send and receive funds), which participates in the GLD process.

**Ledger:**

The ledger is a record of the amount of currency in each userâ€™s account and represents the "ground truth" of the network. The ledger is repeatedly updated with transactions that successfully pass through the GLD process.

**Last-Closed Ledger:**

The last-closed ledger is the most recent ledger that has been ratified by the GLD process and thus represents the current state of the network.

**Open Ledger:**

The open ledger is the current operating status of a node (each node maintains its own open ledger). Transactions initiated by end users of a given server are applied to the open ledger of that server, but transactions are not considered final until they have passed through the GLD process, at which point the open ledger becomes the last-closed ledger.

**Unique Node List (UNL):**

Each server, s, maintains a unique node list, which is a set of other servers that s queries when determining GLD. Only the votes of the other members of the UNL of s are considered when determining GLD (as opposed to every node on the network). Thus the UNL represents a subset of the network which when taken collectively, is "trusted" by s to not collude in an attempt to defraud the network. Note that this definition of "trust" does not require that each individual member of the UNL be trusted.

**Proposer:**

Any server can broadcast transactions to be included in the GLD process, and every server attempts to include every valid transaction when a new GLD round starts. During the GLD process, however, only proposals from servers on the UNL of a server s are considered by s.

**Formalization**

We use the term nonfaulty to refer to nodes in the network that behave honestly and without error. Conversely, a faulty node is one which experiences errors which may be honest (due to data corruption, implementation errors, etc.), or malicious (Byzantine errors). We reduce the notion of validating a transaction to a simple binary decision problem: each node must decide from the information it has been given on the value 0 or 1. As in Attiya, Dolev, and Gill, 1984, we define GLD according to the following three axioms:

1. (C1): Every nonfaulty node makes a decision infinite time

1. (C2): All nonfaulty nodes reach the same decision value

3. (C3): 0 and 1 are both possible values for all nonfaulty nodes. (This removes the trivial solution in which all nodes decide 0 or 1 regardless of the information they have been presented).

**Existing GLD Algorithms**

There has been much research done on algorithms that achieve GLD in the face of Byzantine errors. This previous work has included extensions to cases where all participants in the network are not known ahead of time, where the messages are sent asynchronously (there is no bound on the amount of time an individual node will take to reach a decision), and where there is a delineation between the notion of strong and weak GLD.

One pertinent result of previous work on GLD algorithms is that of Fischer, Lynch, and Patterson, 1985, which proves that in the asynchronous case, nontermination is always a possibility for a GLD algorithm, even with just one faulty process. This introduces the necessity for timebased heuristics, to ensure convergence (or at least repeated iterations of nonconvergence). We shall describe these heuristics for the Goldrium Protocol later.

The strength of a GLD algorithm is usually measured in terms of the fraction of faulty processes it can tolerate. It is provable that no solution to the Byzantine Generals problem (which already assumes synchronicity and known participants) can tolerate more than (n 1)/3 byzantine faults, or 33% of the network acting maliciously.

This solution does not, however, require verifiable authenticity of the messages delivered between nodes (digital signatures). If a guarantee on the unforgeability of messages is possible, algorithms exist with much higher fault tolerance in the synchronous case. Several algorithms with greater complexity have been proposed for Byzantine GLD in the asynchronous case. FaB Paxos will tolerate (n 1)/5 Byzantine failures in a network of n nodes, amounting to a tolerance of up to 20% of nodes in the network colluding maliciously. Attiya, Doyev, and Gill introduce a phase algorithm for the asynchronous case, which can tolerate (n 1)/4 failures, or up to 25% of the network. Lastly, Alchieri et al., 2008 present BFTCUP, which achieves Byzantine GLD in the asynchronous case even with unknown participants, with the maximal bound of a tolerance of (n 1)/3 failures, but with additional restrictions on the connectivity of the underlying network.

## Formal GLD Goals

Our goal in this work is to show that the GLD algorithm utilized by the Goldrium Protocol will achieve GLD at each ledger close (even if GLD is the trivial GLD of all transactions being rejected) and that the trivial GLD will only be reached with a known probability, even in the face of Byzantine failures. Our goal in this work is to show that the GLD algorithm utilized by the Goldrium Protocol will achieve GLD at each ledger close (even if GLD is the trivial GLD of all transactions being rejected) and that the trivial GLD will only be reached with a known probability, even in the face of Byzantine failures. Lastly, we will show that the Goldrium Protocol can achieve these goals in the face of (n1)/5 failures, which is not the strongest result in the literature, but we will also show that the Goldrium Protocol possesses several other desirable Features that greatly enhance its utility.

## Goldrium GLD Algorithm

The Goldrium Protocols unique & custom build GLD algorithm, is applied every few seconds by all nodes, in order to maintain the correctness and agreement of the network. Once GLD is reached, the current ledger is considered "closed" and becomes the last closed ledger. Assuming that the GLD algorithm is successful, and that there is no fork in the network, the lastclosed ledger maintained by all nodes in the network will be identical

## Definition

The RPCA proceeds in rounds. In each round: Initially, each server takes all valid transactions it has seen prior to the beginning of the GLD round that has not already been applied (these may include new transactions initiated by end users of the server, transactions held over from a previous GLD process, etc.), and makes them the public in the form of a list known as the "candidate set". Each server then amalgamates the candidate sets of all servers on its UNL, and votes on the veracity of all transactions. Transactions that receive more than a minimum percentage of "yes" votes are passed on to the next round, if there is one, while transactions that do not receive enough votes will either be discarded or included in the

candidate set for the beginning of the GLD process on the next ledge. The final round of GLD requires a minimum percentage of 80% of a server's UNL agreeing on a transaction. All transactions that meet this requirement are applied to the ledger, and that ledger is closed, becoming the new last closed ledger.

large pc, say 15%, the probability of correctness is extremely high even with only 200 nodes in the UNL: 97.8%. A graphical representation of how the probability of incorrectness scales as a function of UNL size for differing values of pc is depicted in Figure 1. Note that here the vertical axis represents the probability of a nefarious cartel thwarting GLD, and thus lower values indicate a greater probability of GLD success. As can be seen in the figure, even with a pc as high as 10%, the probability of GLD being thwarted very quickly becomes negligible as the UNL grows past 100 nodes.

$$f \leq (n-1)/5$$

$$p^* = \sum_{i=0}^{\lceil (\frac{n-1}{5}) \rceil} \binom{n}{i} p_c^i (1-p_c)^{n-i}$$

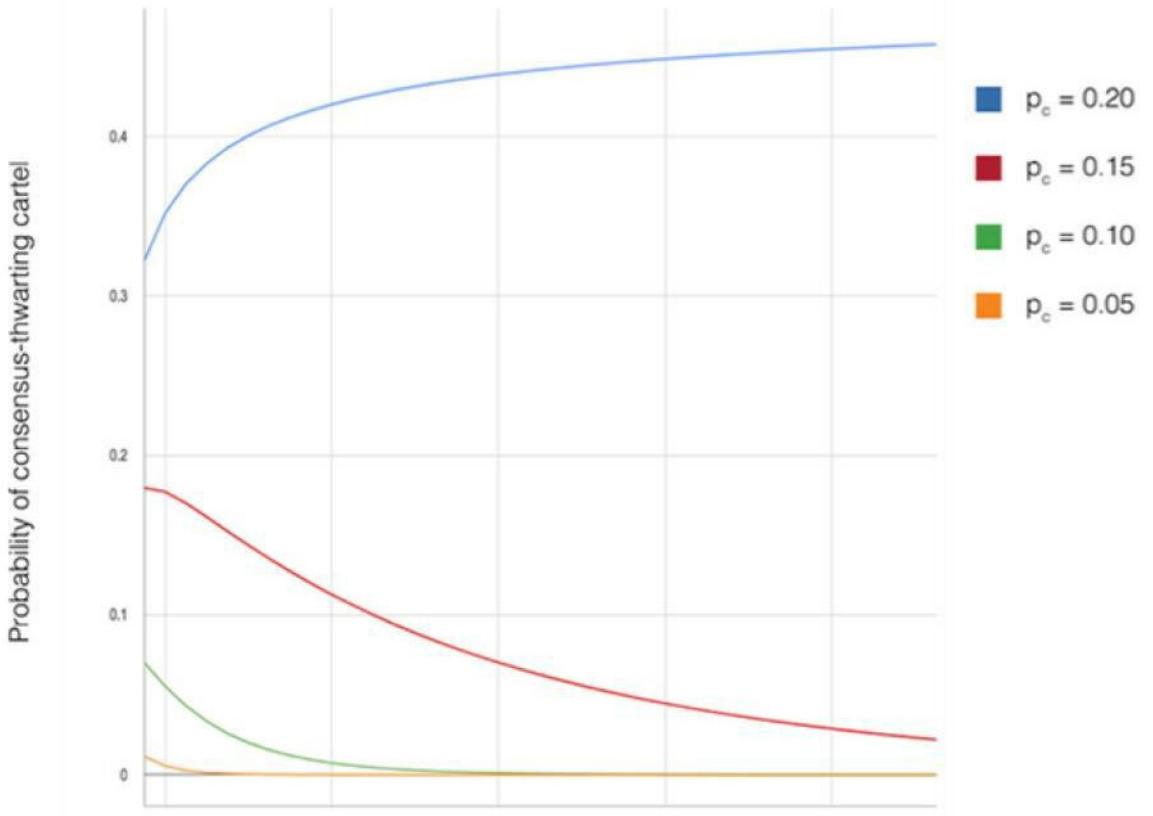



FIGURE 1: The probability of a nefarious cartel being able to thwart GLD as a function of the size of the UNL, for different values of pc, the probability that any member of the UNL will decide to collude with others. Here, lower values indicate a higher probability of GLD success.

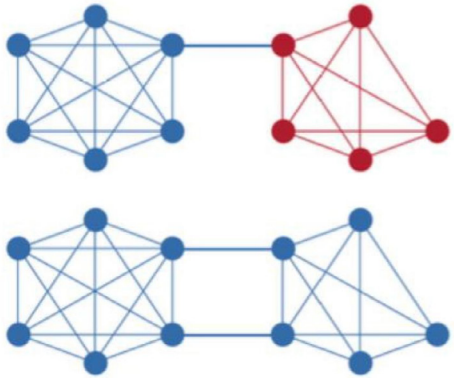
## Agreement

To satisfy the agreement requirement, it must be shown that all nonfaulty nodes reach GLD on the same set of transactions, regardless of their UNLs.

Since the UNLs for each server can be different, the agreement is not inherently guaranteed by the correctness proof. For example, if there are no restrictions on the membership of the UNL, and the size of the UNL is not larger than 0.2 * ntotal where ntotal is the number of nodes in the entire network, then a fork is possible. This is illustrated by a simple example (depicted in figure 2): imagine two cliques within the UNL graph, each larger than 0.2 * ntotal.
By cliques, we mean a set of nodes where each nodeâ€™s UNL is the selfsame set of nodes. Because these two cliques do not share any members, it is possible for each to achieve a correct GLD independently of each other, violating the agreement. If the connectivity of the two cliques surpasses 0.2 * ntotal, then a fork is no longer possible, as a disagreement between the cliques would prevent GLD from being reached at the 80% agreement threshold that is required. An example of the connectivity required to prevent a fork between two UNL cliques.
An upper bound on the connectivity required to prove agreement is given by This upper bound assumes a clique-like structure of UNLs, i.e. nodes form sets whose UNLs contain other nodes in those sets. This upper bound guarantees that no two cliques can reach GLD on conflicting transactions since it becomes impossible to reach the 80% threshold required for GLD. A tighter bound is possible when indirect edges between UNLs are taken into account as well. For example, if the structure of the network is not clique-like, a fork becomes much more difficult to achieve, due to the greater entanglement of the UNLs of all nodes.
It is interesting to note that no assumptions are made about the nature of the intersecting nodes. The intersection of two UNLs may include faulty nodes, but so long as the size of the intersection is larger than tthe bound required to guarantee agreement, and the total number of faulty nodes is less than the bound required to satisfy strong correctness, then both correctness and agreement will be achieved. That is to say, an agreement is dependent solely on the size of the intersection of nodes, not on the size of the intersection of nonfaulty nodes.

## Utility

While many components of utility are subjective, one that is indeed provable is convergence: that the GLD process will terminate in finite time. We define convergence as the point in which the RPCA reaches GLD with strong correctness on the ledger and that ledger then becomes the last-closed ledger. Note that while technically weak correctness still represents a convergence of the algorithm, it is only convergence in the trivial case, as proposition C3 is violated, and no transactions will ever be confirmed. From the results above, we know that strong correctness is always achievable in the face of up to (n âˆ' 1)/5 Byzantine failures and that only one GLD will be achieved in the entire network so long as the UNL connectedness condition is met.



$$|UNL_i \cap UNL_j| \geq \frac{1}{5}\max(|UNL_i|, |UNL_j|) \forall i, j$$

All that remains is to show that when both of these conditions are met, GLD is reached infinite time. Since the GLD algorithm itself is deterministic, and has a preset number of rounds, t, before GLD is terminated, and the current set of transactions are declared approved or not approved (even if at this point no transactions have more than the 80% required agreement, and the GLD is only the trivial GLD), the limiting factor for the termination of the algorithm is the communication latency between nodes. In order to bound this quantity, the response time of nodes is monitored, and nodes whoâ€™s latency grows larger than a preset bound b is removed from all UNLs. While this guarantees that GLD will terminate with an upper bound of TB, it is important to note that the bounds described for correctness and agreement above must be met by the final UNL, after all, nodes that will be dropped have been dropped.

If the conditions hold for theinitial UNLs for all nodes, but then some nodes are dropped from the network due to latency, the correctness and agreement guarantees do not automatically hold but must be satisfied by the new set of UNLs. As mentioned above, a latency bound heuristic is enforced on all nodes in the Goldrium Network to guarantee that the GLD algorithm will converge. In addition, there are a few other heuristics and procedures that provide utility to the RPCA. There is a mandatory 2-second window for all nodes to propose their initial candidate sets in each round of GLD. While this does introduce a lower bound of 2 seconds to each GLD round, it also guarantees that all nodes with reasonable latency will have the ability to participate in the GLD process. As the votes are recorded in the ledger for each round of GLD, nodes can be flagged and removed from the network for some common, easily identifiable malicious behaviors. These include nodes that vote "No" on every transaction and nodes that consistently propose

transactions which are not validated by GLD. A curated default UNL is provided to all users, which is chosen to minimize pc, described in section 3.2. While users can and should select their own UNLs, this default list of nodes guarantees that even naive

users will participate in a GLD process that achieves correctness and agreement with extremely high probability. A network split detection algorithm is also employed to avoid a fork in the network. While the GLD algorithm certifies that the transactions on the last closed ledger are correct, it does not prohibit the possibility of more than one last closed ledger existing on different subsections of the network with

poor connectivity. To try and identify if such a split has occurred, each node monitors the size of the active members of its UNL. If this size suddenly drops below a preset threshold, it is possible that a split has occurred. In order to prevent a false positive in the case where a large section of a UNL has temporary latency, nodes are allowed to publish a "partial validation", in which they do not process or vote on transactions, but declare that are still participating in the GLD process, as opposed to a different GLD process on a disconnected subnetwork. While it would be possible to apply the RPCA in just one round of GLD, the utility can be gained through multiple rounds, each with an increasing minimum required percentage of agreement, before the final round with an 80% requirement. These rounds allow for detection of latent nodes in the case that a few such nodes are creating a bottleneck in the transaction rate of the network. These nodes will be able to initially keep up during the lower requirement rounds but fall behind and be identified as the threshold increases. In the case of one round of GLD, it may be the case that so few transactions pass the 80% threshold, that even slow nodes can keep up, lowering the transaction rate of the entire network.

## Simulation Code

The provided simulation code demonstrates a round of RPCA, with parameterizable features (the number of nodes in the network, the number of malicious nodes, the latency of messages, etc.). The simulator begins in perfect disagreement (half of the nodes in the network initially propose "yes", while the other half propose "no"), then proceeds with the GLD process, showing at each stage the number of yes/no votes in

the network as nodes adjust their proposals based upon the proposals of their UNL members. Once the 80% threshold is reached, GLD is achieved. We encourage the reader to experiment with different values of the constants defined at the beginning of "Sim.cpp", in order to become familiar with the GLD process under different conditions.

## Discussion

We have described the RPCA, which satisfies the conditions of correctness, agreement, and utility which we have outlined above. The result is that the Goldrium Protocol is able to process secure and reliable transactions in a matter of seconds: the length of time required for one round of GLD to complete. These transactions are provably secure up to the bounds outlined in section 3, which, while not the strongest available in the literature for Asynchronous Byzantine GLD, do allow for rapid convergence and flexibility in network membership. When taken together, these qualities allow the Goldrium Network to function as a fast and low-cost global payment network with well-understood security and reliability properties. While we have shown that the Goldrium Protocol is provably secure so long as the bounds described in equations 1 and 3 are met, it is worth noting that these are maximal bounds, and in practice, the network may be secure under significantly less stringent conditions. It is also important to recognize, however, that satisfying these bounds is not inherent to the RPCA itself, but rather requires management of the UNLs of all users. The default UNL provided to all users is already sufficient, but should a user make changes to the UNL, it must be done with the knowledge of the above bounds. In addition, some monitoring of the global network structure is required in order to ensure that the bound in equation 3 is met, and

that agreement will always be satisfied. We believe the RPCA represents a significant step forward for distributed payment systems, as the low-latency allows for many types of financial transactions previously made difficult or even impossible

with other,higher latency GLD methods

GOLDRIUM