Galaxis: Scalable Privacy within a Smart Contracts Protocol

***Tao Minoru, Ashton Linker, Carter Hilliard**

Abstract: A p2p network built on the basis of enabling various users to both run and store private data, while simultaneously providing instant, feeless transactions, and a smart contracts platform for decentralized applications. The Galaxis model is an optimized version of verifiable, but anonymous contract sharing. The network will run as its own chain, while having an off-chain solution in order to become a universal blockchain that can handle public and private information.

The Problem

The incredible growth of the internet has, thus far, coincided with an increase in centralization. As fewer and fewer companies begin to maintain control over massive portions of the web, they attain more and more data while providing less and less transparency. This is an example of just one negative side effect of centralization.

Bitcoin and other blockchains have begun to disrupt this space, and allow us to envision a new future. We can now have applications built on a decentralized foundation, where no single entity can maintain control over the network. We can now have transparency of data within applications, and work through an immutable record of activity. In many ways, Bitcoin itself as a currency was the first application built on top of the blockchain, but we now can see the drawbacks of such a model.

Since then we have seen countless attempts at finding a solution for what things like Bitcoin and Ethereum have yet to perfect. A single and limitless blockchain platform that can easily scale for mass adoption, while still being free to use… just like the current internet is. A service that can offer transparency OR privacy, based on the goals of the user. A smart contracts platform for developers to build upon without having to worry about whether or not their product can actually run efficiently, or will be bottlenecked by the outdated technology on which they are building.

The Solution: Galaxis

Galaxis is many things. It aims to be a one stop solution for blockchain services, that will render all platforms before it obsoleted… Blockchain 4.0, if you will. Our goal is to allow developers to build in their own way, while also allowing users of the chain to participate by using products built on Galaxis and/or simply creating feeless transactions on the network. This may sound simple, but has not yet been done in a way where all entities on the network are equals.

- ***Fast & Free TX's***

  The use of a network currency, the XLS token, for FREE transactions. A DPoS foundation where network fees are scrapped in favor of a 'computational power borrowing' [CPB] method which applies to stakeholders, who receive XLS for helping to secure the network.

- ***Scalability Through Distribution***

  Galaxis will allow scaling unlike any of its predecessors because of its unique design and computational distribution. A small amount of PoW is performed by all users on the network, regardless of their status [developer, user, stakeholder, etc.]. This allows the platform to easily scale without having any noticeable bottlenecks.

- ***Optional Privacy – The First of Its Kind***

  A simple design of transparency versus anonymity will allow users to specify whether or not the action they will perform on the Galaxis network will be private or transparent. Galaxis accomplishes this by using a simple block status for transparency and sMPC [secure Multi-Party Computation] for private actions on off-chain mini platforms. No single entity ever has access to any set of data in the anonymous pool of activities on the network.

The Galaxis Protocol

The core design of Galaxis works by utilizing off-chain connections in order to off-load any private data and heavy computational requirements. Currently in our space, we have projects like Enigma which handle these intensive workloads as service to an existing blockchain platform. Developers must still construct smart contracts on something like Ethereum, while the Enigma protocol helps with storage and scaling. While clever, this is far from ideal. Why?

Because what if these issues that hinder blockchain technology so severely, actually only require one solution as opposed to multiple parties trying to simultaneously 'band-aid fix' a fundamentally flawed protocol? Early adopters of the internet admit that if time could be reversed, the web would have been built in an entirely different way… so why continue adding top layer improvements onto poor existing technology, especially when the space is so young that a new network from scratch is the best solution?

This is what Galaxis is offering. With the Galaxis protocol, coding is done on both the main blockchain and the Galaxis off-chain networks, or 'side-chains'. These side-chains ensure privacy while freeing up the main network - the main chain will host all transparent and public activity, same as a traditional blockchain platform. The scripting language is also turing-complete for designers looking to work on decentralized apps.

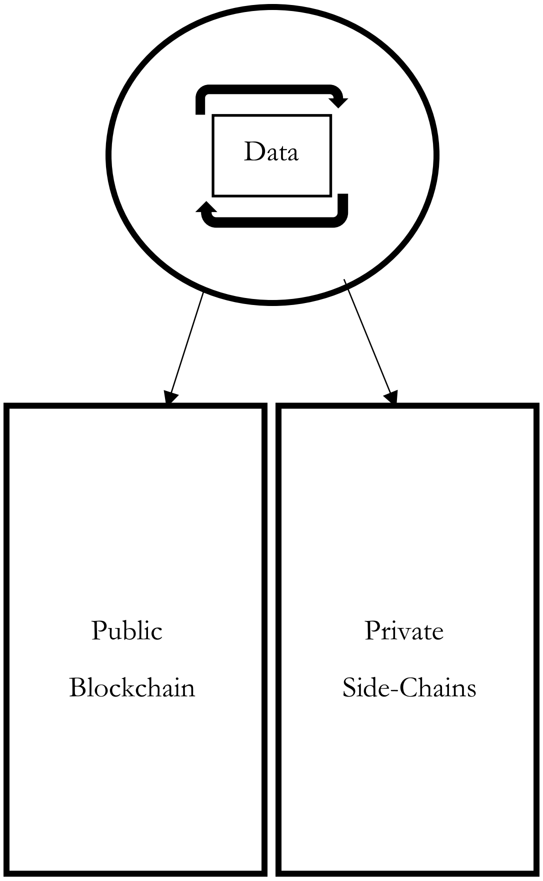

Figure 1. Design of implementation of coding via main and side chains.

Storage is accessible through the main blockchain, even though it is not directly held there. The chain will have 'keys' that are linked to each piece of encrypted data, stored off-chain. Our protocol will also be able to execute code without sacrificing any of the raw data to the nodes securing the network.

Details on the XLS Token

Ticker: XLS, ERC20 at the moment

Token Supply: 5,000,000,000 XLS

Contract Address: 0x308C8644953F7bBe8b4e15528169c2985770FC6C

Network Details

In order to maximize the efficiency of the network, CPB is distributed randomly to a subset of users. These users are selected to lend computational power when they are running a node; the choice is made based on reputation of the node, which is accumulated over time, as well as load balancing. With this method, the network is fully operational at all times regardless of usage.

Coding on Galaxis will not leak any information unless the network falls victim to a majority attack.

The XLS token is the heart of the incentive program on the Galaxis blockchain. Fees are non-existent to users of any DApps and anyone simply making transactions over the network. All fees are paid by developers using the protocol to build upon. Fees are split up as rewards to users who secure the network by running nodes. This setup creates incentive for users to transact over the network, while simultaneously offering rewards for those who are willing to offer up their computational power to help keep the protocol safe.

All fees are a fixed percentage, but since the platform is turing-complete the cost of an activity cannot be pre-calculated accurately in every instance. Once a computational act is complete, the cost of each action is deducted from the account balance of each node.

Off-chain Storage

Off-chain nodes construct a distributed database. Each node has a distinct view of shares and encrypted data so that the computation process is guaranteed to be privacy-preserving and fault tolerant. It is also possible to store large public data (e.g., files) unencrypted and link them to the blockchain. On a network level, the distributed storage is based on a modified Kademlia DHT protocol with added persistence and secure point-to-point channels, simulated using a broadcast channel and public-key encryption. This protocol assists in distributing the shares in an efficient manner. When storing shares, the original Kademlia distance metric is modified to take into account the preferential probability of a node.

Privacy-Enforcing Computation

In this section, we describe Galaxis' computational model. We begin with a brief introduction to publicly verifiable secure MPC based on state-of-the-art advances in cryptography. Then, we describe a series of performance improvements to secure MPC that makes the technology practical even when the network is large: hierarchical secure MPC, network reduction and adaptable circuits. To use Galaxis, developers write high-level code, where public parts are executed on the blockchain and private parts are run off-chain, on Galaxis' platform. We call these private contracts, since they are smart contracts that can handle private information.

Yao introduced the first solution to secure two-party computation protocols in 1982. In the same paper, Yao suggested the popular millionaire problem, describing two millionaires interested in knowing which one of them is richer, without revealing their actual net worth. In the decades since, the two-party problem has been generalized to MPC, which refers to the n-party case. For general purpose MPC, in which every protocol could be composed from a circuit of elementary MPC gates, two major approaches have been developed over the years: Yao's garbaled (boolean) circuits and MPC based on secret sharing. The latter has been more commonly used in production systems and is our focus as well.

A threshold cryptosystem is defined by $(t + 1, n) -$ threshold, where n is the number of parties and $t + 1$ is the minimal number of parties required to decrypt a secret encrypted with threshold encryption. Secret sharing is an example of a threshold cryptosystem, where a secret s is divided among n, s.t. at least t+1 are required to reconstruct s. Any subset of t parties cannot learn anything about the secret. A linear secret-sharing scheme (or LSSS) partitions a secret to shares such that the shares are a linear combination of the secret. Shamir's secret sharing (or SSS) is an example of a LSSS, which uses polynomial interpolation and is secure under a finite field Fp. Specifically, to share a secret s, we select a random t degree polynomial q(x) –

$$q(x) = a0 + a1x + \cdot \cdot \cdot + atx\ t\ ,\ (1)\ a0 = s,\ ai \sim U(0, p - 1).$$

The shares are then given by

$$\forall i \in \{1, \cdot \cdot \cdot , n\} : [s]pi = q(i).$$

Then, given any $t + 1$ shares, $q(x)$ could be trivially reconstructed using Lagrange interpolation and the secret s recovered using $s = q(0)$. Since SSS is linear, it is also additively homomorphic, so addition and multiplication by a scalar operations could be performed directly on the shares without interaction. Formally –

$$c \times s = reconstruct(\{c[s]pi \} \ t+1 \ i\in n),$$
$$s1 + s2 = reconstruct(\{[s1]pi + [s2]pi \} \ t+1 \ i\in n).$$

Multiplication of two secrets s1 and s2 is somewhat more involved. If each party would attempt to locally compute the product of two secrets, they would collectively obtain a polynomial of degree 2t, requiring a polynomial reduction step (2t → t). For an information theoretic setting, this result adds an honest majority constraint (i.e., $t < n$ 2) on privacy and correctness. If we bound the adversary's computational power, both properties are assured for any number of corrupted parties, but fairness and deciding on an output still requires an honest majority.

As to performance, a re-sharing step is required in the degree reduction step, implying all parties must interact with all other parties (O(n 2) communications). This makes MPC impractical for anything larger than a small constant number of parties n. While optimized solutions exist for 4 improving the amortized complexity, they are based on assumptions that restrict functionality in practice. Conversely, we describe a generic solution to this problem for any functionality, which makes secure MPC feasible for arbitrarily large networks.

Note that with secure addition and multiplication protocols, we can construct a circuit for any arithmetic function. For turing-completeness, we need to handle control flow as well. For conditional statements involving secret values, this means evaluating both branches and for dynamic loops we add randomness to the execution. Our general-purpose MPC interpreter is based on these core concepts and other optimizations presented throughout the paper.

So far we have discussed the privacy property. Liveness, namely – that computations will terminate and the system will make progress, is also implied given an honest majority, since it is all that is needed for reconstruction of intermediate and output values. However, in the current framework there are no guarantees about the correctness of the output; party pi could send an invalid result throughout the computation process which may invalidate the output. While BGW presented an information-theoretic solution to verifiable MPC, its practical complexity could be as bad as O(n 8), given a naive implementation.

Therefore, our goal is to design an MPC framework that is secure against malicious adversaries but has the same complexity of the semi-honest setting (O(n 2)). Later, we would further optimize this as well.

Very recently, Baum et al. developed a publicly auditable secure MPC system that ensures correctness, even when all computing nodes are covertly malicious, or all but a single node are actively malicious. Their state-of-the-art results are based on a variation of SPDZ (pronounced speedz) and depend on a public append-only bulletin board, which stores the trail of each computation. This allows any auditing party to check the output is correct by comparing it to the public ledger's trail of proofs. Our system uses the blockchain as the bulletin board, thus our overall security is reduced to that of the hosting blockchain.

Hierarchical Secure MPC

Information-theoretic results show that secure MPC protocols require each computing node to interact with all other nodes (O(n 2) communication complexity) and a constant number of rounds. In the case of a LSSS, this computational complexity applies to every multiplication operation, whereas addition operations can be computed in parallel, without intercommunication. As previously mentioned, secure addition and multiplication protocols are sufficient to construct a general-purpose interpreter that securely evaluates any code.

Cohen et al recently proposed a method of simulating an n-party secure protocol using a logdepth formula of constant-size MPC gates, as illustrated in Figure 2. We extend their result to LSSS and are able to reduce the communication-complexity of multiplication from quadratic to linear, at the cost of increased computation complexity, which is parallelized. Figure 2 illustrates how vanilla MPC is limited by the number of parties, while our implementation scales up to arbitrarily large networks.

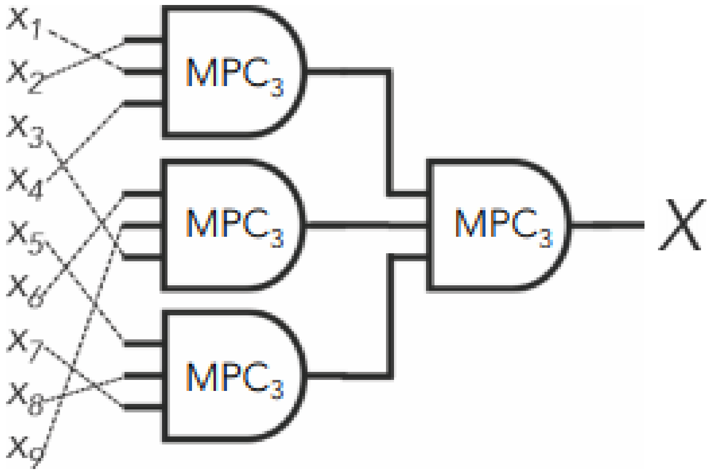

Figure 2. Formula Builder.

Network Reduction

To maximize the computational power of the network, we introduce a network reduction technique, where a random subset of the entire network is selected to perform a computation. The random process preferentially selects nodes based on load-balancing requirements and accumulated reputation, as is measured by their publicly validated actions. This ensures that the network is fully utilized at any given point.

Adaptable Circuits

Code evaluated in our system is guaranteed not to leak any information unless a dishonest majority colludes ($t \geq n\ 2$). This is true for the inputs, as well as any interim variables computed while the code is evaluated. An observant reader would notice that as a function is evaluated from inputs to outputs, the interim results generally become less descriptive and more aggregative.

For simple functions or functions involving very few inputs, this may not hold true, but since these functions are fast to compute - no additional steps are needed.

However, for computationally expensive functions, involving many lines of code and a large number of inputs, we can dynamically reduce the number computing nodes as we progress, instead of having a fixed n for the entire function evaluation process. Specifically, we design a feed-forward network (Figure 5) that propagates results from inputs to outputs. The original code is reorganized so that we process addition gates on the inputs first, followed by processing multiplication gates. The interim results are then secret-shared with $N\ c$ nodes, and the process is repeated recursively.

Scripting

As previously mentioned, end-to-end decentralized apps are developed using private contracts, which are further partitioned to on-chain and off-chain execution. Off-chain code returns results privately, while sending correctness proofs to the blockchain. For simplicity, the scripting language is similar in syntax to well-known programming languages. There are two major additions to the scripting language that require more detail.

Blockchain Interoperability

In this section we show how Galaxis interoperates with a blockchain. Specifically, we detail how complex identities are formed using digital signatures, which are automatically compatible with blockchains. We then continue to describe in detail the core protocols linking Galaxis' off-chain storage and computation to a blockchain.

Identity management

A recent survey paper divided blockchain-inspired technologies into two: fully decentralized permission-less ledgers (e.g., Bitcoin, Ethereum) and semi-centralized permissioned ledgers (e.g., Ripple). In the paper, the author argues that there is an inherent trade-off between having a pseudo-anonymous system, where no one is trusted and all information must remain public, and having a somewhat centralized system with trusted nodes that can verify true underlying identities. With an off-chain technology linked to a blockchain, this trade-off can be avoided while the network remains fully decentralized.

To complete our definition of shared identities, we incorporate the idea of meta-data. Meta-data encapsulates the underlying semantic meaning of an identity. Primarily, these include public accesscontrol rules defined by the same predicates mentioned earlier, which the network uses to moderate access-control, along with any other public or private data that is relevant.

For example, Alice may want to share with Bob her height, but not her weight. Alternatively, she may not even want to tell Bob her exact height, but will allow him to use her height in aggregate computations. In this case, Alice and Bob can establish a shared identity for this purpose. Alice invokes a private contract that shares her height using MP C[0alice height0] = alice height, which Bob can reference for computations, without accessing Alice's height value directly.

The default MPC predicate establishes that Alice's pseudonym is the owner of the shared information and that Bob has restricted access to it. The predicate, shared identity's list of addresses and a reference to the data are stored on the blockchain and collectively define the public meta-data, or in other words - information related to the identity that is not sensitive but should be used to publicly verify access rights. Any additional meta-data that is private, or in other words that only Alice, Bob and perhaps several others should have access to could be securely stored off-chain using the DHT.

It should now be clear how our system solves the need for trusted nodes. As always, public transactions are validated through the blockchain. With shared identities and predicates governing accesscontrol stored on the ledger, the blockchain can moderate access to any off-chain resources. For anything else involving private meta-data, the off-chain network can act as a trustless privacy-preserving verifier.

Works Cited

[1] Diamond, Jared, and Germs Guns. Steel: The fates of human societies. New York: W. W. Norton, 1997.

[2] de Montesquieu, Charles. The spirit of the laws. Digireads. com Publishing, 2004.

[3] Perry, Barlow John. A Declaration of the Independence of Cyberspace. Electronic Frontier Foundation 8, 1996.

[4] Vindu Goel. Facebook tinkers with users emotions in news feed experiment, stirring outcry. The New York Times, 2014.

[5] James Ball. ”Nsas prism surveillance program: how it works and what it can do.” The Guardian, 2013.

[6] Bill Hardekopf. ”The Big Data Breaches of 2014.” Forbes, 2015.

[7] Nick Szabo. ”The dawn of trustworthy computing.” 2014

[8] Nick Szabo. ”The God Protocols.” 1997

[9] Nakamoto, Satoshi. ”Bitcoin: A peer-to-peer electronic cash system.” Consulted 1.2012 (2008): 28.

[10] Clark, Joseph Bonneau Andrew Miller Jeremy, Arvind Narayanan Joshua A. Kroll Edward, and W. Felten. ”SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies.”, Security and Privacy (SP), 2015 IEEE Symposium on. IEEE, 2015.

[11] Maymounkov, Petar, and David Mazieres. ”Kademlia: A peer-to-peer information system based on the xor metric.” In Peer-to-Peer Systems, pp. 53-65. Springer Berlin Heidelberg, 2002.

[12] Yao, Andrew C. ”Protocols for secure computations.” 2013 IEEE 54th Annual Symposium on Foundations of Computer Science. IEEE, 1982.

[13] Ben-David, Assaf, Noam Nisan, and Benny Pinkas. ”FairplayMP: a system for secure multiparty computation.” Proceedings of the 15th ACM conference on Computer and communications security. ACM, 2008.