xx network

White Paper
xx cMix

V2.0

April, 2021

xx.network

The information provided in this document pertaining to Privategrity Corporation **or xx Labs SECZ** (”**xx network**” or the ”Company”), the xx Coin (the “Coins”), its business assets, strategy and operations is for general informational purposes only and is not a formal offer to sell or a solicitation of an offer to buy any Coins, securities, options, futures, or other derivatives related to securities in any jurisdiction and its content is not prescribed by securities laws. Information contained in this document should not be relied upon as advice to buy or sell or hold Coins or securities or as an offer to sell Coins. This document does not take into account nor does it provide any tax, legal or investment advice or opinion regarding the specific investment objectives or financial situation of any person. While the information in this presentation is believed to be accurate and reliable, **xx network** and its agents, advisors, directors, officers, employees and shareholders make no representation or warranties, expressed or implied, as to the accuracy of such information and **xx network** expressly disclaims any and all liability that may be based on such information or errors or omissions thereof. **xx network** reserves the right to amend or replace the information contained herein, in part or entirely, at any time, and undertakes no obligation to provide the recipient with access to the amended information or to notify the recipient thereof.

Neither we nor any of our representatives shall have any liability whatsoever, under contract, tort, trust or otherwise, to you or any person resulting from the use of the information in this presentation by you or any of your representatives or for omissions from the information in this presentation. Additionally, the Company undertakes no obligation to comment on the expectations of, or statements made by, third parties in respect of the matters discussed in this presentation.

This document contains forward looking statements, including among other things, statements concerning the distribution of xx Coins, and other statements identified by words such as “could,” “expects,” “intends,” “may,” “plans,” “potential,” “should,” “will,” “would,” or similar expressions and the negatives of those terms. Forward-looking statements are not promises or guarantees of future performance, and are subject to a variety of risks and uncertainties, many of which are beyond our control. Actual results could differ materially from those anticipated in such forward-looking statements as a result of various risks and uncertainties, which include, without limitation, market risks and uncertainties and the satisfaction of losing conditions for a distribution of xx coins. Forward-looking statements speak only as of the date hereof, and, except as required by law, **xx network** undertakes no obligation to update or revise these forward-looking statements.

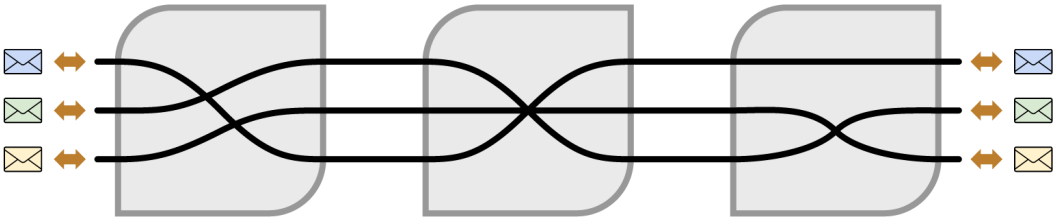
Introduction

As the internet has become increasingly pervasive, many aspects of our lives have been digitized and recorded as data by centralized entities. Economic incentives to exploit this data as well as an increase in massive data breaches have resulted in a huge uptick in privacy violations. To protect consumers, some platforms promise end-to-end encryption to limit access to message content. But this is not enough; greater protection is needed for each user's **metadata**. Metadata consists of the *who*, *what*, *when*, *where*, and *how* details of any message or activity.

To address these concerns, xx network has developed a privacy-protecting communication layer through a decentralized implementation of cMix, a mixnet protocol for anonymous communications. The team behind the creation of xx cMix is composed of pioneers who developed early practical, anonymous, and verifiable cryptographic systems. Its members are among the first to propose and deploy digital currencies, mix networks, unpermissioned cryptography, verifiable voting systems, and many other advances in cryptography. By understanding how cMix synthesizes their prior work, readers can evaluate the opportunity that xx network offers to dApp developers, node operators, and consumers seeking unparalleled privacy, and gain an understanding of the cMix privacy protecting layer within the larger xx network framework.

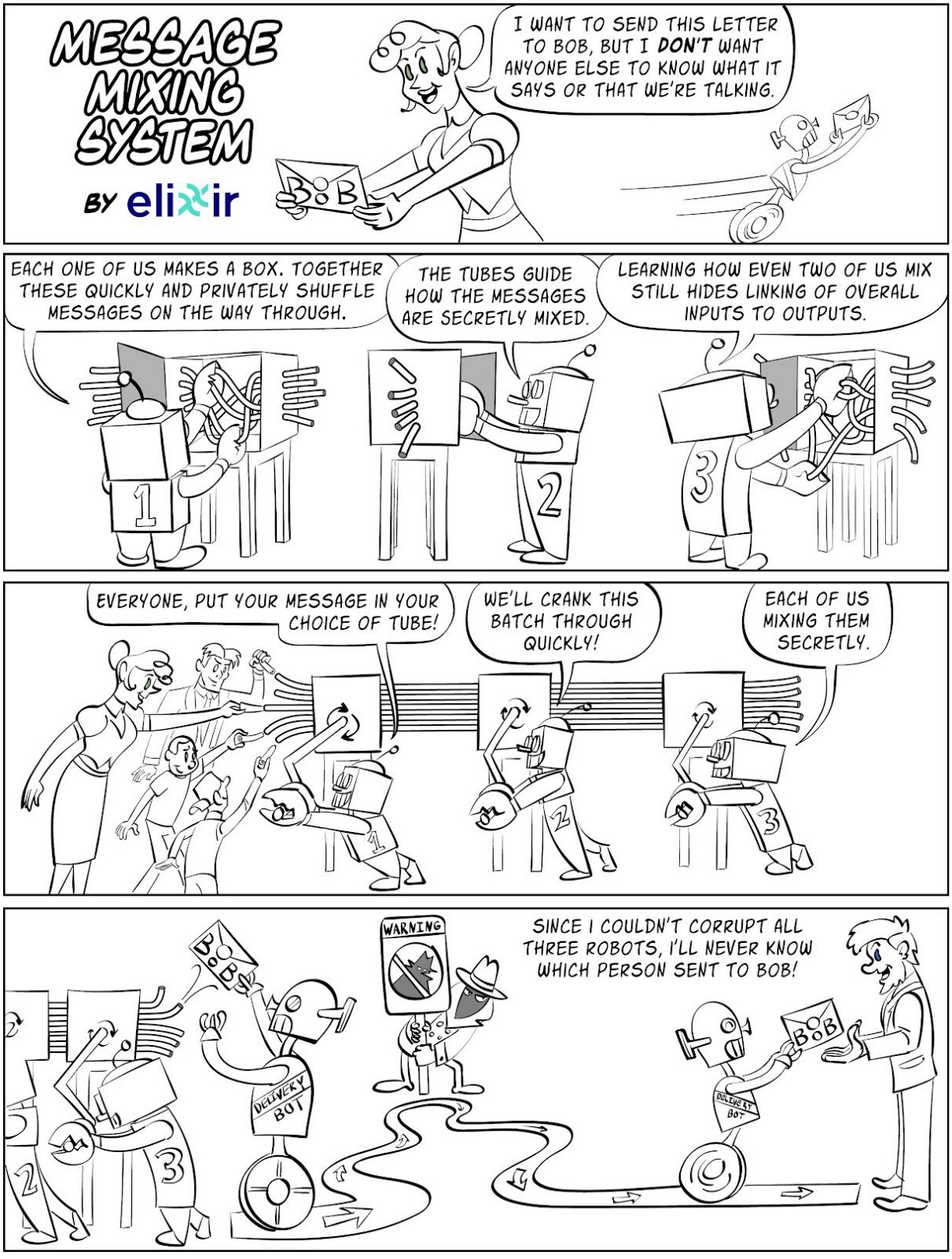
Metadata Protection

xx network is working to provide metadata protection by implementing a variant of the cMix protocol at consumer scale, fulfilling two core values fundamental to achieving true security and privacy. The first value is **confidentiality**—protecting the identity of participants in activities, such as a message sender and recipient. This means that an adversary cannot map any input to the corresponding output with any higher probability than random guessing, even if the adversary has compromised most of the system. The second value is **integrity**—verifying the trustworthiness of the transaction system. This means that at any given point, either the cMix system successfully delivers all messages without alteration, or, in the event of a failure, any malicious mix-node should be identifiable with overwhelming probability.



To achieve both anonymity and integrity, the cMix system brings together two key concepts: mixnets and precomputation. **Mixnets**, also known as mixing networks, were first described by xx network Founder David Chaum in 1979. A mixnet lays down cryptographic rules for messages or transaction activity from a set of users to be relayed by a sequence of trusted intermediaries known as mix-nodes. These mix-nodes receive a batch of encrypted messages, randomly permute or “mix” them, and then send them forward.

The main drawback to traditional mixnets is that the public key operations securing the mixing operation are time-consuming when performed at scale, making mixnets too slow for most consumer uses. cMix aims to solve this problem by using **precomputation**. Precomputation allows mix-nodes to do the time-consuming public key cryptography before the real-time phase of handling messages between senders and recipients. The result is a very efficient type of mixnet that allows users to send and receive messages in real-time without compromising security or privacy.


MESSAGE MIXING SYSTEM
BY elixir
I WANT TO SEND THIS LETTER TO BOB, BUT I DON'T WANT ANYONE ELSE TO KNOW WHAT IT SAYS OR THAT WE'RE TALKING.
BOB
EACH ONE OF US MAKES A BOX. TOGETHER THESE QUICKLY AND PRIVATELY SHUFFLE MESSAGES ON THE WAY THROUGH.
THE TUBES GUIDE HOW THE MESSAGES ARE SECRETLY MIXED.
LEARNING HOW EVEN TWO OF US MIX STILL HIDES LINKING OF OVERALL INPUTS TO OUTPUTS.
1
2
3
EVERYONE, PUT YOUR MESSAGE IN YOUR CHOICE OF TUBE!
WE'LL CRANK THIS BATCH THROUGH QUICKLY!
EACH OF US MIXING THEM SECRETLY.
WARNING
DELIVERY BOT
SINCE I COULDN'T CORRUPT ALL THREE ROBOTS, I'LL NEVER KNOW WHICH PERSON SENT TO BOB!

Scalable Architecture

Mix-nodes perform the work of decrypting messages and mixing traffic to hide the associations between senders and recipients. The work of xx network's variant of cMix is performed in two phases: *precomputation*, and *real-time*. In the precomputation phase, the mix-nodes establish shared values to circumvent the need for public key operations during the real-time phase. In the real-time phase, mix-nodes receive messages, perform the encryption and decryption work prepared during precomputation, and pass the message on to the next mix-node.

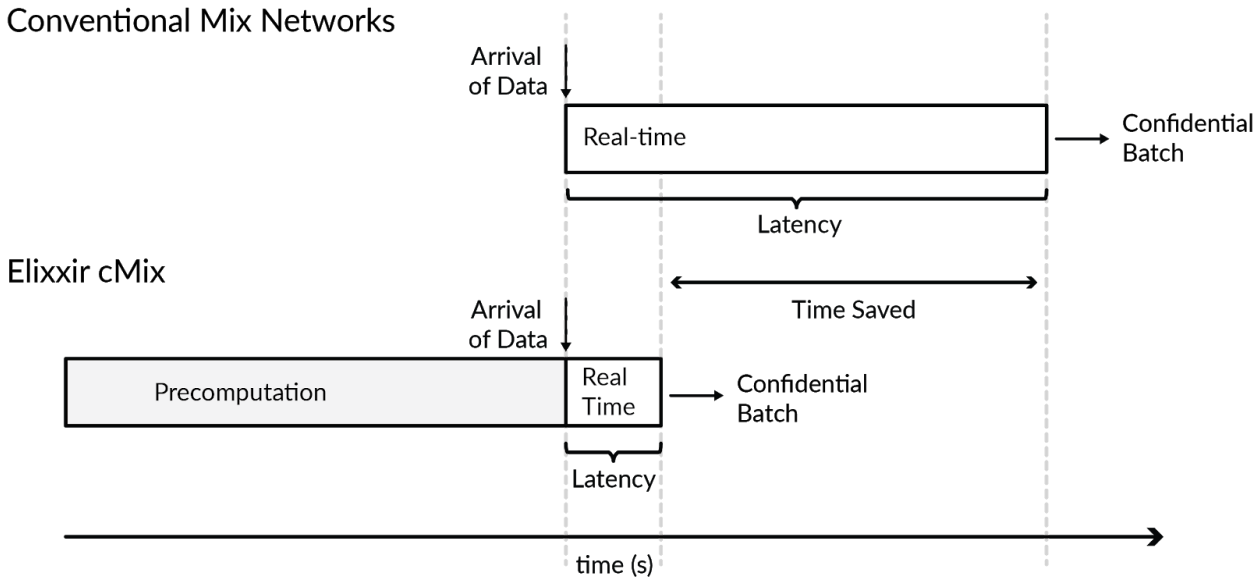
Teams

In xx network, unlike other decentralized protocols, groups of nodes are unmanipulatably organized into small ephemeral **Teams**. cMixx teams are temporary and only exist to mix a single batch of messages. The teams are formed in an unmanipulatable manner by *xx consensus*. Teams independently agree to process all messages beforehand, then mix and decrypt[1] them, delivering the results to the recipients and payments to *xx consensus*. Following completion, the team disbands and member nodes become available to be placed on a new random team.

Precomputation

All operations conducted by teams are accelerated through the use of **Precomputation**. These precomputations produce a template that dictates how the nodes within a team must process information during mixing. Consequently, the template is completely defined before the batch of messages arrives. The use of precomputation ensures confidentiality while dramatically increasing the speed at which information can be processed.

[1] Messages are still end-to-end encrypted

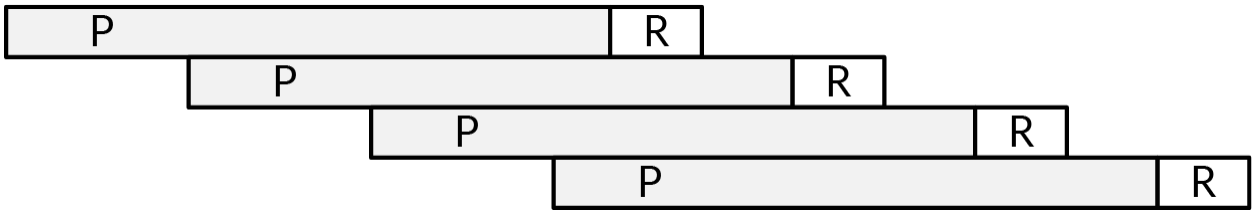


Time-consuming, computationally intensive team precomputation is performed before transactions are sent or received. This allows very fast real-time computation of transactions to process each batch of messages. The use of precomputation decouples security from latency as seen in conventional mix networks, delivering a greater level of security and privacy without the latency penalty.

There are two phases involved in batch mixing by a team as depicted in the figure above. First, the team performs a computationally intensive precomputation, producing a unique template defining how the information or messages will be processed. When the batch of messages are received, the nodes of the team work together to process the messages in real time according to this unique template—a process that takes less than 1/20th of the precomputation time.

Scaling

At any given time, tens, hundreds, or even thousands of teams will exist within the network in varying stages of precomputation. However, only one team will be mixing messages. These precomputations overlap, as shown below.



Teams are organized into a pipeline to maximize the number of transactions that can be processed by our platform.

As nodes are added to the network and the number of teams increases, each team has longer intervals between team formation and engaging in real-time message processing. This allows the team to dedicate the increased time to preparing larger precomputations capable of processing more messages, thereby scaling up the throughput of the network as a whole. In fact, as nodes are added to the network, the throughput increases roughly proportionally. This is a much sought-after scaling property known as **linear scalability**.

Mixing Messages and Transactions

Each team runs a single instance of a mixing network (mixnet) based on the cMix protocol). Besides supporting secure communications between users, these features may be leveraged and expanded to provide key functionality for *the xx consensus* protocol.

The cMix protocol is itself a breakthrough: it exhibits drastically lower real-time cryptographic latency than any other mixnet due to its use of precomputation. In xx network's implementation, real-time message processing in cMix involves three operations: **reception**, **permutation**, and **delivery**.

- **Reception:** Commutative network encryption based on modular group arithmetic is added to each already end-to-end encrypted message. Simultaneously, user-to-network encryption is removed from it, disassociating the sender's identity from the message.
- **Permutation:** The order of messages within a batch is shuffled, removing an observer's ability to correlate the order in which messages were received with the senders' identities.
- **Delivery:** The remaining network encryption is canceled using the cryptographic outputs of precomputation, revealing the destination and the end-to-end encrypted messages.

The precomputation operation mimics the real-time operation, executing fundamentally the same steps with a null input to produce just the total encryption for each slot as added in the real time phase. In order to not leak intermediary keys, these operations are executed under partially homomorphic encryption based on ElGamal. Therefore, in addition to the three operations, a fourth operation is added to precomputation to strip the homomorphic encryption and produce the cryptographic outputs used in the delivery phase.

The cMix Protocol has two additional important features that make it unique:

1. **Return Path:** The return path allows a receiver to send an immediate response through the mixnet; this permits receipts of transactions to be returned to users without the platform needing to know addressing information, thereby hiding the identity of the sender. To accomplish this goal, nodes generate additional keying material for the return path and apply an inverse permutation so that responses are received by the original senders.

2. **Commitments:** Commitments are a protocol that produces data, often produced through hashing, that allows a third party to audit a computation performed by a node later. All messages exchanged between nodes, the permutations they perform, and all keying materials in the mixnet's

precomputation inherently function as commitments of how messages will be processed in the future, during the real-time phase of block generation. Nodes also produce a commit of the batch of encrypted messages before any decryption takes place. Commits function as an efficient mechanism for verifying that nodes perform their operations correctly.

With these features, xx network provides integrity and anonymity for users sending messages and transactions through the platform. Any honest node can, with non-negligible probability, identify nodes that violate integrity and prevent malicious nodes from improperly framing an honest node. Lastly, any single honest node in a team is able to protect user anonymity.

Conclusion and Further Reading

For further details on cMix, the academic paper[2] (first published in 2016 dives into the design and features of cMix. It also describes the early progress in developing cMix: the conceptual work, considerations about adversaries and how to defeat them, and the results of a proof-of-concept test. As of April 2021, xx network has implemented its variant of cMix and has successfully deployed the xx network Public BetaNet. The xx network will next roll out the MainNet, with the intention to bring security and privacy to consumers with unprecedented speed and scale.

[2] David Chaum et al. “cMix: Mixing with Minimal Real-Time Asymmetric Cryptographic Operations.” In: ACNS . Ed. by Dieter Gollmann, Atsuko Miyaji, and Hiroaki Kikuchi. Vol. 10355. Lecture Notes in Computer Science. Springer, 2017, pp. 557–578. ISBN: 978-3-319-61204-1. URL: https: / /dblp.uni-trier.de/db/conf/acns/acns2017.html#conf/acns/ChaumDJKKRS17
.