cMixx$^+$

David Chaum[1] and Mario Yaksetig[2]

[1] xx labs
david@chaum.com
[2] BitFashioned SEZC
info@bitfashioned.com

Abstract. We introduce cMixx$^+$, an efficient way to use a re-encryption to “mix” distinctive series of codes that can then be queried to see if a specific distinctive code is included in a set of mixed outputs. This functionality helps, for example, owners of a digital currency that is maintained on a central bank ledger (CBDC) to spend their money and have the return communication loop (that prevents a double spend) go through cMixx$^+$ so that where and when that spending happened is hidden from a network adversary.

Keywords: Anonymous communications · mix network · privacy.

1 Introduction

Presently, in the digital age, ensuring the privacy of online communications is a critical challenge. cMixx, the implementation of cMix [4] on the xx network [3], introduces a novel privacy solution that allows for a secure and anonymous transmission of data packets. cMixx operates by batching data packets together and mixing them through a unique cascade of five randomly selected mix nodes. This process makes it virtually impossible for a network adversary to link any input message to an output message. Consequently, the connection between the sender and the receiver is effectively anonymized.

cMixx provides the ability to handle various types of data payloads, suitable for messaging, photo sharing, and even complex commercial transactions. This wide feature set makes cMixx a versatile tool.

Beyond message transmission, there is a growing need to securely protect the exchange of cryptographic keys and codes (e.g., iris codes [2]). Extending on the foundational work from Golle et al. [5], cMixx$^+$ enhances this protection by introducing a new re-encryption approach that allows for secure verification of inclusion of these codes within an encrypted set. This feature is particularly useful in contexts like Central Bank Digital Currencies (e.g., Tourbillon [1]), as well as in secure identity authentication processes.

To maintain its decentralized and secure nature, cMixx$^+$ is supported by a network of decentralized nodes, akin to the original cMixx. This ensures the system remains resilient against centralization and control by any single entity.

2 Protocol Overview

The core idea of the protocol is that there is a set of (decentralized) mix nodes that actively receive input batches and continuously perform re-encryption operations to the corresponding batch of encrypted codes (or random serial numbers). As a result, each node outputs a batch where each encrypted element is unlinkable to the corresponding input elements.

The set of nodes allow for an entry to be privately queried. Therefore, upon receiving this lookup request, the nodes re-encrypt the entire batch to be encrypted under a same verifiable key. The user performing the lookup then uploads the encrypted random number to be queried. The mix nodes re-encrypt this value to be encrypted under the same key as the entire batch. The resulting ciphertext should match an index in the batch. If not, the item is not present.

3 System Model & Architecture

We now describe the system entities and associated roles, the architecture of the system, and the adversarial model.

3.1 System Entities and Roles

In our design we consider the following entities: users, message relayers (or senders), mix nodes, and receivers.

3.2 Architecture Diagram

Users, denoted as $\mathcal{U}$, connect to a sender, denoted as $\mathcal{S}$. This sender acts as an entry point to the mix network. The mix network is comprised by n nodes, each responsible for performing a re-encryption operation to the received input batches. The output is then relayed to a receiver, denoted as $\mathcal{R}$. This receiver can be instantiated as a ledger. We show in Figure. 1 the corresponding architecture.

4 Protocol Description

Our protocol consists of two parts: insertion and lookup.

4.1 Insertion

The goal of this protocol is to allow a user to insert an item into a database in an encrypted manner. To do so, the user inserts an item into the mix cascade. The ongoing batch effectively acts as a re-randomizable (encrypted) database. We show in Figure 2 a more detailed version of this protocol.

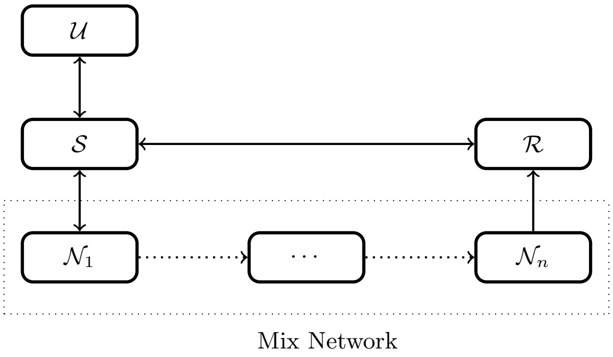

Fig. 1. cMixx$^+$ System Architecture.

4.2 Lookup

The goal of this protocol is to allow a user to lookup whether or not an item is present in the encrypted database. This is done by submitting a lookup request to the mix cascade. The mix cascade, upon receiving this request, performs a collective decryption and re-encrypts the mix batch using one same key for every element. This encryption is performed in a verifiable manner. The user then submits the encrypted random number to look up. The mix applies the same set of operations and determines if there is a match on the output. We show in Figure 3 a more detailed version of this protocol.

5 Conclusion

The cMixx network and its advanced iteration, cMixx Plus, represent a significant advancement in the field of digital communication privacy and security. Through innovative mix network design and efficient handling of data packets, these systems address the growing concerns of privacy and anonymity in the digital space.

cMixx, with its unique approach to precomputing the cryptographic operations before batching and shuffling data packets through a network of randomly selected nodes, establishes a new paradigm in obscuring the trail of communication, effectively severing the link between sender and receiver in a very efficient manner. Its capacity to handle diverse data payloads makes it a versatile tool for various applications, ranging from personal communication to complex commercial transactions.

The evolution into cMixx Plus marks a further enhancement in privacy technology. By introducing an efficient re-encryption method to mix distinct codes within a random number series, cMixx Plus expands its utility to more sophisticated cryptographic applications. This feature is invaluable in contexts where privacy in the validation of cryptographic keys and codes is paramount, such as in digital currency transactions and identity authentication processes.

The decentralized structure of both cMixx and cMixx Plus ensures a robust defense against centralized control, aligning with the foundational principles of privacy and security in the digital world.

In conclusion, the cMixx network systems demonstrate a forward-thinking approach to digital privacy and security. They provide a blueprint for future developments in the field and set a high standard for privacy-oriented communication technologies. The implications of these systems extend beyond their immediate applications, offering insights and frameworks that can guide the development of secure digital communication tools in an increasingly interconnected world.

References

1. Project tourbillon. https://www.bis.org/about/bisih/topics/cbdc/tourbillon.htm, accessed: November 16, 2023
2. What is an iris code, and how does it preserve privacy? https://worldcoin.org/blog/worldcoin/what-is-iris-code-how-does-it-preserve-privacy, accessed: November 16, 2023
3. xx network. http://xx.network, accessed: November 16, 2023
4. Chaum, D., Das, D., Javani, F., Kate, A., Krasnova, A., de Ruiter, J., Sherman, A.T.: cmix: Mixing with minimal real-time asymmetric cryptographic operations. Cryptology ePrint Archive, Paper 2016/008 (2016), https://eprint.iacr.org/2016/008
5. Golle, P., Jakobsson, M., Juels, A., Syverson, P.F.: Universal re-encryption for mixnets. In: Okamoto, T. (ed.) Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings. Lecture Notes in Computer Science, vol. 2964, pp. 163–178. Springer (2004)

A Insertion Protocol Description

Withdrawal Protocol

Random Value Generation - User generates the value and prepares the payload. We assume that mix has a public-key pair $(x, y = g^x)$. To perform the generation process, the user performs the following steps:

- Generates a random value: $x \xleftarrow{\$} \{0,1\}^\lambda$
- (ElGamal) Encrypts the value using the mix public key: $c = (g^r, m \times y^r)$
- Generates a random pair of encryption factors: $(k_0, k_1) \xleftarrow{\$} \{0,1\}^\lambda$
- Generates the first part of the payload: $\alpha_0 = g^{k_0}$
- Generates the second part of the payload: $\beta_0 = m \times y^{k_0}$
- Generates the third part of the payload: $\alpha_1 = g^{k_1}$
- Generates the fourth part of the payload: $\beta_1 = y^{k_1}$
- Forms final ciphertext payload: $[(\alpha_0, \beta_0), (\alpha_1, \beta_1)]$

Mixing - Mix node receives an input batch of ciphertexts and "re-encrypts" it such that the outputs are mixed and unlinkable to the inputs. To perform this mixing step, each mix nodes perform the following steps for each of the input ciphertexts:

- Generates new re-encryption factor: $(k'_0, k'_1) \xleftarrow{\$} \{0,1\}^\lambda$
- Calculate $\alpha'_0 = \alpha_0 \alpha_1^{k'_0}$
- Calculate $\beta'_0 = \beta_0 \beta_1^{k'_0}$
- Calculate $\alpha'_1 = \alpha_1^{k'_1}$
- Calculate $\beta'_0 = \beta_1^{k'_1}$
- Permute input ciphertext index to a new one in the output batch
- Output $[(\alpha'_0, \beta'_0), (\alpha'_1, \beta'_1)]$

Fig. 2. Insertion Protocol.

B Lookup Protocol Description

Lookup Protocol

Lookup request - Party $\mathcal{R}$ receives an image check. The goal of this step is to allow for a party to check whether or not a specific element is included in the mixed batch.

- Party triggers the image check protocol

Re-encryption - In this step, the mix nodes re-encrypt the payload to a specific value. To do so, the mix nodes perform the following steps:

- Decrypt all the ciphertexts and re-encrypt them to encrypt elements to a different verifiable key. For example, a different key z
- (Optional) Produce a zero-knowledge proof π that each payload is properly formed (i.e., the message is raised to the correct exponent)

Query

- User uploads the hash image to the nodes
- Nodes collectively work together to re-encrypt payload to make it encrypted with the z value
- Nodes output encrypted version of hash image along with proof of correct encryption.
- User can check if image is on the output list
- Item is removed from the batch

Fig. 3. Payment Protocol.