MiXiM: Mixnet design decisions and empirical evaluation

Iness Ben Guirat
imec-COSIC, KU Leuven
ibenguir@esat.kuleuven.be

Devashish Gosain
Max Planck Institute for Informatics
dgosain@mpi-inf.mpg.de

Claudia Diaz
imec-COSIC, KU Leuven
claudia.diaz@esat.kuleuven.be

ABSTRACT

In this paper we present MiXiM, a simulation framework for mixnets that allows researchers to evaluate different design options and their tradeoffs. This framework is flexible and allows to quickly run experiments to assess combinations of mixnet building blocks, such as mixing strategies and network topologies, as well as study the effect of different parameters related to each component. The framework provides results for a number of metrics including anonymity, end-to-end latency and traffic overhead.

CCS CONCEPTS

• **Security and privacy**;

KEYWORDS

Mixnet, Simulation, Empirical evaluation

1 INTRODUCTION

In the last decades, a variety of overlay networks [2, 12, 15, 18, 20] have been proposed to provide communication anonymity, meaning that in these networks it is not possible to find out who is communicating with whom. Mixnets are a type of anonymous communication network that aims to be secure against *global network adversaries*, who observe all communications in the underlying network. Even though the concept of mixnets [1] predates onion routing [10] by more than a decade, and early mixnet deployments [3, 16] were operative before Tor, their uptake has remained far behind for years, mainly due to their higher computational requirements, added latency, and lack of industrial-quality implementations. In recent years however a number of mixnet designs have been proposed [12, 15, 18] and currently the Nym network is implementing a mixnet-based anonymity network already deployed as a testnet prototype [6].

Like other types of anonymity networks, mixnets are complex systems with many components and parameters, including mixing strategies, routing policies, and dummy traffic strategies. Questions like—what is the best combination of mixing strategies and topologies, or what is the impact of the number of mixes per layer on anonymity, *etc.* have not been explored systematically.

Each of these components and its parametrization can have a significant impact on the anonymity provided by a network. In order to select components and tune the parameters of a mixnet configuration, designers need to be able to evaluate the anonymity resulting from possible design decisions in different conditions. Simulation is commonly used in research studies on the Tor network, particularly in instances where the experiments that need to be conducted would endanger actual users if they were deployed in the live network. Instead, many attacks on Tor can be evaluated with Shadow [13] in a safe manner. Shadow provides an isolated environment for simulating Tor network communications. It produces traces and logs that can be further analyzed to assess what an adversary would be able to learn with access to different subsets of the logged data.

In this paper we introduce MiXiM, a generic mixnet simulation framework to evaluate anonymity in different designs and configurations. MiXiM's level of abstraction focuses on elements core to anonymous routing, including mixnet topology, routing policy, mixing algorithms, cover traffic and mix corruption. We focus on documenting the metadata such as packets sources, destinations and timings exposed by the mixnet, while making abstraction of data payloads and cryptographic operations. In addition, MiXiM captures a number of relevant metrics (latency, bandwidth overhead etc.) that can be used to study tradeoffs between anonymity, performance and cost. The simulator and all related tooling are publicly available.[1]

2 BACKGROUND AND RELATED WORK

A *mixnet* is an overlay network of **mixes**. Mixes are servers that cryptographically transform and reorder messages, such that inputs are unlinkable to outputs both in terms of message appearance and timing—even by an adversary who can monitor *all* the messages going in and out of the mix. Message reordering can be achieved with different "mixing" strategies [8]. MiXiM presently supports four popular algorithms:

- *Threshold Mix:* A threshold mix [1] buffers messages in the queue until a set number T (threshold parameter) is reached. At that point the mix permutes the T messages, flushing them in a random order.
- *Timed Mix:* A timed mix buffers messages in the queue for a set time interval (timeout). When the time is elapsed, the mix permutes the messages it has collected in the interval, flushing them in a random order.
- *Pool Mix:* These are a variation of both *threshold* and *timed* mixes where, instead of flushing all messages, a fraction of messages is kept [3]. MiXim supports pool mixes but they are left out of the experiments presented here due to space limitations.
- *Stop-and-Go Mix:* a SG-mix [14] processes messages individually and continuously in time. Each received message is kept

[1] https://gitlab.esat.kuleuven.be/Iness.BenGuirat/mixim

WPES '21, November 15, 2021, Virtual Event, Republic of Korea

ACM ISBN 978-1-4503-8527-5/21/11…$15.00
https://doi.org/10.1145/3463676.3485613

for a random amount of time (sampled from an exponential distribution) and sent out when the time has elapsed. Due to the memory-less properties of exponential distributions, individual delays result in a reordering of the sequence of messages that has high entropy [4].

Mixnet topology is the arrangement that defines how mixes are inter-connected in the mixnet and thus which routes (sequences of mixes) exist for messages to follow.

- *Cascade:* In this topology, all messages traverse a fixed sequence of mixes forming a chain with a predetermined order: each mix sends messages to the next mix in the cascade [1]. A single cascade cannot scale to handle more throughput than that of a single mix. Thus, if more capacity is required, multiple cascades can run in parallel [2, 11].
- *Stratified:* In stratified or *layered* topologies mixes are arranged in a fixed number of layers where each mix, at any given time, is assigned to only one specific layer. The layers are interconnected such that each mix in layer i receives messages from mixes in layer $i-1$ and sends messages to mixes in layer $i+1$. There are two variants of stratified topology, (i) Fully connected where mixes in layer i can send messages to all mixes in layer $i+1$, and (ii) Not Fully connected where mixes in layer i can send messages to only a subset of mixes in layer $i+1$.

3 MIXIM FRAMEWORK

The MiXiM framework consists of configuration files that define the mixnet environment, a discrete-event simulator that instantiates and executes the network, producing observations that are logged to files, and analysis scripts to process the log files and extract empirical results.

3.1 Configuration

Configuration files define the topology of the mixnet, routing policy, client and network traffic characteristics, and optional features such as cover traffic and mix corruption. These configuration values are input to the simulator that instantiates the network and its components (as processes) and then simulates its execution by generating and processing events.

Client information: Sets the total number of clients C, the message generation rate per client λ_C, and the duration of the simulation.

Mixing information: Sets the mix types with their parameters *e.g.,* μ for Poisson mix, *timeout* for Timed mix, and *threshold* for Threshold mix.

Topology information: Sets the topology with its parameters *e.g.,* number of layers, the number of mixes per layer, the number of cascades, *etc.*

Optional information: If desired, a system designer can also generate dummy traffic. In MiXiM, dummy traffic is generated according to a Poisson process with parameter λ_D. In addition, MiXiM supports mix corruption where α is the fraction of corrupted mixes. Table 1 provides a handy reference for all the parameters.

C	Total number of clients
λ_C	Mean rate of message generation per client
mix-type	Timed, threshold, pool, or Poisson
T_0	Timeout between mix flushes (Timed mix)
T	Threshold parameter (Threshold mix)
μ	Mean message delay (Poisson mix)
net-type	Cascades, Stratified Fully Connected (FC) or Restricted
l	Path length
N_M	Number of mixes
λ_D	Mean rate of dummy traffic per mix
α	Fraction of corrupted mixes

Table 1: Summary of parameters.

3.2 Implementation

The MiXiM simulator is a discrete event simulator. It is built with SimPy[2] and models all the building blocks of a mixnet as described in §2. The components (mixes, clients, and messages) are processes that exist in the same environment. Once the mixnet parameters are chosen, MiXiM loads the configuration file and instantiates the simulation environment where all processes live in. They interact with the environment and with each other via *events*.

The simulator instantiates the mixnet as follows: first it creates the *Network* process according to the configuration, including the number of mixes, the topology, the types of mixes and the number and positions of corrupt mixes. Then MiXiM instantiates each of the *Mixes* with their specific descriptions (types and parameters). After this step, *Client* processes are created that in turn generate and send *Messages*. We model client message sending behaviour as a Poisson process[3] with the parameter λ_C. MiXiM can be extended to implement alternative distributions to model other client sending behaviours that may be of interest.

The simulator runs for the specified time and then computes the evaluation metrics from the log files. Since MiXiM is a discrete event simulator, simulation time and "wall-clock" time do not progress in-step. We denote the simulation time unit of SimPy as t_u. Furthermore, upon start-up the network requires a **burn-in** period time to become **stable**. Stability means that each mix in the mixnet is receiving and transmitting the expected number of messages following the specified parameters in the configurations files.

MiXiM evaluates the anonymity of messages that are transiting the network once it is stable. The simulator logs all relevant information (probability distributions over messages, latencies, etc.) to files. The final stage of the workflow is the analysis step where scripts[4] parse these log files to compute the results. We use entropy as metric for anonymity[9, 19] (ref. App. A for details).

4 MIXNET BUILDING BLOCKS

In this section, we demonstrate the usefulness of the simulator by walking through a rather simple yet complete mixnet-based system assessing the different parameters and design decisions.

[2] A python discrete event simulation library, https://simpy.readthedocs.io/en/latest/

[3] Poisson processes are extensively used to model natural phenomena, such as user message sending habits.

[4] The framework provides scripts for all the analysis conducted in this paper. However, the logging feature is extensible and can allow additional evaluations with new scripts.

4.1 Mixing

In this experiment we study the impact of different mixing strategies on anonymity (ref. §2 for more details on mixing strategies).

Experimental setup: We set the average end-to-end latency L_e = 1s, the number of clients $C = 100$ and two values of λ_C *i.e.,* $\lambda_C = 1$ and $\lambda_C = 10$. In a Poisson mix $L_e = 1$s translates to $\mu = 1$, while in a timed mix it corresponds to a timeout $T_0 = 2$s. Given the considered message rates λ_C, we adjust the threshold parameter T to achieve $L_e = 1$s and perform a fair comparison. This results in the following parameter values:

For $C \cdot \lambda_C = 100$: $\mu = 1$, $T_0 = 2$s, $T = 200$.

For $C \cdot \lambda_C = 1000$: $\mu = 1$, $T_0 = 2$s, $T = 2000$.

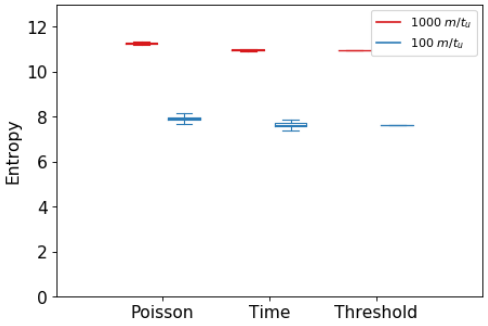

Figure 1: Impact of mixing strategies on Entropy.

Result: In Figure 1 it is evident that the three types of mixing provide approximately the same entropy, with a slight advantage for Poisson mixing.

4.2 Topologies

Mix cascades are a classical design [1] that has been used in many proposals, such as XRD [15], Vuvuzela [12], and cMix [2]. Stratified topologies have been recommended as optimal topologies for anonymous routing in prior studies [7] and are used in systems such as Loopix [18]. Although XRD [15] and Loopix [18] seem different, they are both designed for scalability, aim at hiding metadata, and are capable of storing messages. The main differences between the two types of designs are (1) topology: Vuvuzela, cMix and XRD's topology consists of organizing the mixes into small chains (cascades) each acting as a local mixnet, while Loopix is a fully connected stratified network; and (2) mix types: Threshold vs Poisson. Inspired by these designs, we experimentally compare three types of topologies *i.e.,* fully connected stratified, not fully connected stratified, and multiple cascades.

Multiple cascades have the drawback of splitting the anonymity set: users who are connected to one cascade are completely distinguishable from users who are connected to another cascade. To avoid this problem, XRD [15] implements a rather sophisticated scheme of cascade selection. The scheme guarantees that every pair of users have at least one cascade in common (ref. App. B for details). Since our aim is to analyze the impact of underlying topology on anonymity, we do not include all the features of Loopix and XRD, and only compare their topologies. For example in XRD, every client connected to N_c cascades, sends $N_c - 1$ dummy messages each time they send a real message. Moreover, every message (either real or dummy) goes through a different chain. This is very expensive in terms of bandwidth and thus we chose to ignore this feature.

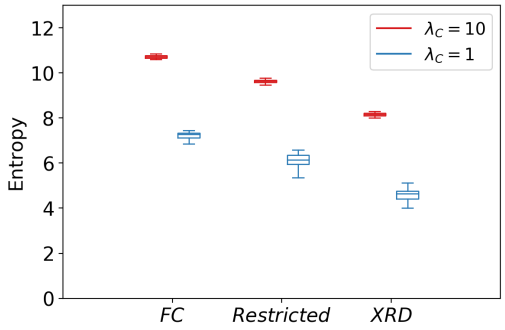

Figure 2: Impact of different topologies on Entropy.

Experimental setup: Since we already studied the impact of different mixing strategies and our aim in this experiment is to analyze the impact of topologies on anonymity, we consider Poisson mixing for all the experiments and only vary the network topology. We evaluate the following network topologies for $C = 100$ and two values of $\lambda_C = 1$ and $\lambda_C = 10$:

- Fully connected (FC) stratified: 3 layers of 6 mixes each.
- Restricted (not fully connected) stratified: 3 layers of 6 mixes each, but each mix is only connected to 2 mixes in the adjacent layers.
- XRD (multiple cascades): 6 cascades of 3 mixes each.

Results: Figure 2 shows that even though we have the same amount of traffic, same mix types and same number of nodes, the network topology greatly influences the level of anonymity: stratified topologies, and in particular fully connected, provide more anonymity than multiple cascades. This is due to the fact that the second layer of mixing aggregates all messages in one large anonymity set, while in cascades the users of each cascade remain partitioned.

4.3 Layers and Average Delay

To build a mixnet based system, another important parameter to decide on is the the number of layers. On one hand adding layers will make the messages mix with each other more therefore anonymity will increase, but on the other hand adding layers increase the risk of packets' loss. Additionally, to increase anonymity one might also opt for increasing the average delay of each message per mix [5].

Experimental setup: In Figure 3, we show results for $C = 100$ with $\lambda_C = 10$, a stratified topology of 10 mixes per layer, with all mixes as Poisson mixes. We study the impact of the number layers l and the average delay for each message per mix μ on anonymity.

Result: Increasing the latency of messages in each mix has a higher impact on anonymity than adding layers. In fact, after 4 layers the entropy barely increases. However going from an average delay of $0.1t_u$ to $10t_u$ per mix in 3 by 10 topology will increase anonymity by more than 6 bits which means an increase of anonymity set size that is $2^6 = 64$-fold.

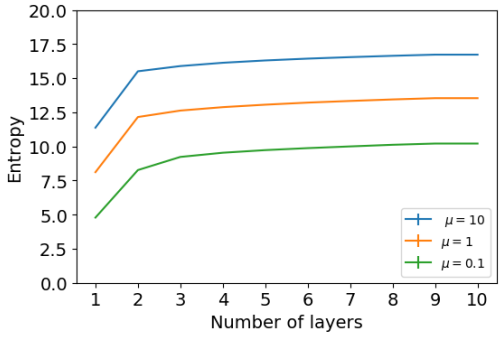

Figure 3: Variation of Entropy depending on the number of layers and the average delay of each message per mix.

4.4 Mix Corruption

We now evaluate anonymity for different fractions mixes that are controlled by the adversary. For these mixes the adversary knows with full certainty the correspondence between inputs and outputs, rather than having probabilistic information on possible correspondences.

Experimental setup: In Figure 4 we show results for a fully connected stratified topology with 10 Poisson mixes per layer with $\mu = 0.1$, and $C = 100$ clients with sending rate $\lambda_C = 10$. We increase the number of layers and study the impact of different fractions of mix corruption on anonymity.

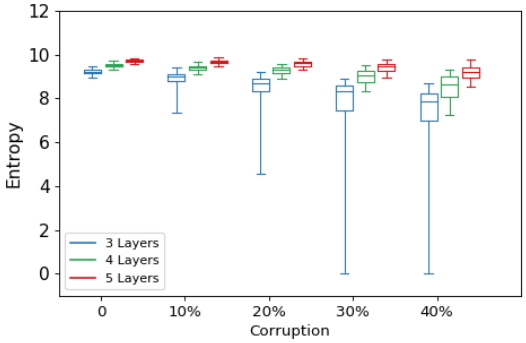

Figure 4: Impact of fraction of corruption (α) on Entropy.

In the previous subsection we showed that, with zero corruption, anonymity does not substantially increase after 3 layers. However, when considering mix corruption, we observe that more layers results in better anonymity. For instance in Figure 4, for 40% corruption, anonymity in a three layer topology has a high standard deviation compared to a four or five layer topology. This is because the chances of a message traversing a fully corrupted path (*i.e.,* encountering a corrupted mix in each layer) are higher when the number of layers is small. With three layers a non-negligible fraction of messages (6%) traverse a fully corrupted path.

4.5 Cover Traffic

Oya et al. [17] described two types of cover traffic: (i) Client-based dummy traffic and (ii) Mix-based dummy traffic. Since we assume that all messages are cryptographically indistinguishable, client-based dummy traffic is considered the same as real traffic. Therefore any change in the amount of traffic, real or client-based dummy, will impact the anonymity in exactly the same way. Mix-based dummy traffic has a different effect, and next we evaluate its impact on anonymity.

Experimental setup: We evaluate anonymity while varying the rate of client traffic λ_C (for $C = 100$ clients) and mix-based dummy traffic λ_D. In Figure 5 we show results for a 4 (layers) by 10 (mixes) fully connected stratified topology of 40 Poisson mixes with $\mu = 0.1$.

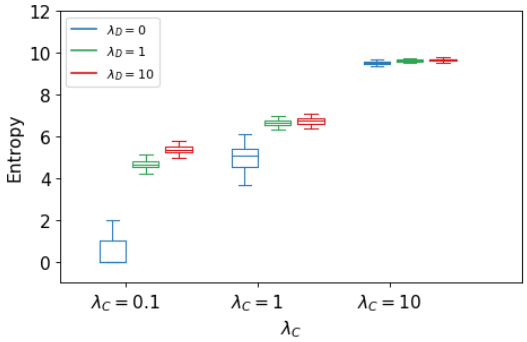

Figure 5: Impact of mix dummy traffic on entropy.

We observe in Figure 5 that dummy traffic significantly increases anonymity when the amount of real traffic is low. *E.g.,* when $\lambda_C = 0.1$ (real traffic is $C \cdot \lambda_C$, *i.e.,* 10 messages/t_u), the median entropy increases from 0 to 5 bits with sufficient dummy traffic. However, when the amount of real traffic is high (when $\lambda_C = 10$) adding dummies does not have any major impact on anonymity.

5 CONCLUSION

Mixnets are anonymous communication networks that provide anonymity against a passive global network adversary. Mixnets have many parameters, *e.g.,* number of layers, types of mixes, underlying topological structure, real and cover traffic rates, number of clients, *etc.* These parameters interplay in complex ways to provide a certain level of anonymity to routed messages.

Thus, in this paper we proposed MiXiM, a framework with which one can i) efficiently understand the interplay of these parameters and, ii) evaluate and design mixnet configurations. MiXiM already supports a variety of design options and configurations that can be easily extended to conduct further studies and perform systematic evaluations of mixnet designs.

Acknowledgments.

We would like to thank Tariq Elahi for early discussions and feedback on this work. This research is partially supported by the Research Council KU Leuven under the grant C24/18/049, by CyberSecurity Research Flanders with reference number VR20192203, and

by DARPA FA8750-19-C-0502. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any of the funders.

REFERENCES

[1] Chaum, D. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM 24*, 2 (1981), 84–88.

[2] Chaum, D., Javani, F., Kate, A., Krasnova, A., Ruiter, J., Sherman, A. T., and Das, D. cmix: Anonymization by high-performance scalable mixing. Tech. rep., Technical report, 2016.

[3] Cottrell, L. Mixmaster and remailer attacks, 1995.

[4] Danezis, G. The traffic analysis of continuous-time mixes. In *International Workshop on Privacy Enhancing Technologies* (2004), Springer, pp. 35–50.

[5] Das, D., Meiser, S., Mohammadi, E., and Kate, A. Anonymity trilemma: Strong anonymity, low bandwidth overhead, low latency-choose two. In *2018 IEEE Symposium on Security and Privacy (SP)* (2018), IEEE, pp. 108–126.

[6] Diaz, C., Halpin, H., and Kiayias, A. The Nym Network. https://nymtech.net/nym-whitepaper.pdf, February 2021.

[7] Diaz, C., Murdoch, S. J., and Troncoso, C. Impact of network topology on anonymity and overhead in low-latency anonymity networks. In *Proceedings of the 10th International Conference on Privacy Enhancing Technologies* (Berlin, Heidelberg, 2010), PETS'10, Springer-Verlag, pp. 184–201.

[8] Diaz, C., and Preneel, B. Taxonomy of mixes and dummy traffic. In *Information Security Management, Education and Privacy*. Springer, 2004, pp. 217–232.

[9] Diaz, C., Seys, S., Claessens, J., and Preneel, B. Towards measuring anonymity. In *Proceedings of the 2nd International Conference on Privacy Enhancing Technologies* (Berlin, Heidelberg, 2002), PET'02, Springer-Verlag, pp. 54–68.

[10] Dingledine, R., Mathewson, N., and Syverson, P. Tor: The second-generation onion router. In *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13* (USA, 2004), SSYM'04, USENIX Association, p. 21.

[11] Gelernter, N., Herzberg, A., and Leibowitz, H. Two cents for strong anonymity: The anonymous post-office protocol. In *International Conference on Cryptology and Network Security* (2017), Springer, pp. 390–412.

[12] Hooff, J. V. D., Lazar, D., Zaharia, M., and Zeldovich, N. Vuvuzela: Scalable private messaging resistant to traffic analysis. In *Proceedings of the 25th Symposium on Operating Systems Principles* (2015), pp. 137–152.

[13] Jansen, R., and Hopper, N. Shadow: Running tor in a box for accurate and efficient experimentation. In *Proceedings of the 19th Symposium on Network and Distributed System Security (NDSS)* (February 2012), Internet Society.

[14] Kesdogan, D., Egner, J., and Büschkes, R. Stop-and-go-mixes providing probabilistic anonymity in an open system. In *International Workshop on Information Hiding* (1998), Springer, pp. 83–98.

[15] Kwon, A., Lu, D., and Devadas, S. {XRD}: Scalable messaging system with cryptographic privacy. In *17th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 20)* (2020), pp. 759–776.

[16] Mathewson, N., and Dingledine, R. Mixminion: Strong anonymity for financial cryptography. In *Proceedings of Financial Cryptography (FC '04)* (February 2004), Springer-Verlag, LNCS 3110, pp. 227–232.

[17] Oya, S., Troncoso, C., and Pérez-González, F. Do dummies pay off? limits of dummy traffic protection in anonymous communications. In *International Symposium on Privacy Enhancing Technologies Symposium* (2014), Springer, pp. 204–223.

[18] Piotrowska, A. M., Hayes, J., Elahi, T., Meiser, S., and Danezis, G. The loopix anonymity system. In *26th {USENIX} Security Symposium ({USENIX} Security 17)* (2017), pp. 1199–1216.

[19] Serjantov, A., and Danezis, G. Towards an information theoretic metric for anonymity. In *Proceedings of the 2nd International Conference on Privacy Enhancing Technologies* (Berlin, Heidelberg, 2002), PET'02, Springer-Verlag, pp. 41–53.

[20] Shirazi, F., Simeonovski, M., Asghar, M. R., Backes, M., and Diaz, C. A survey on routing in anonymous communication protocols. *ACM Computing Surveys (CSUR) 51*, 3 (2018), 1–39.

A ENTROPY

Entropy is an information theoretic measure of the uncertainty associated with a random variable. In anonymous communications, entropy captures the uncertainty of an adversary about which is the target message of interest amongst all of the other messages in the network that it could be confused with. The larger the size of this set and the more uniform the probability distribution of being the target among all the messages, the more anonymous is the target [9, 19].

In MiXiM when all the mixes are stable, the simulator automatically chooses a target message m_t. The simulated adversary starts to track all the M messages in the network, including the target. Because the target message, as well as all the messages in the network, goes through multiple mixes—where they are delayed and shuffled with other messages—the adversary assigns a probability, $0 \leq \Pr[m_i = m_t] \leq 1$, to each message of being the target message, with $i = 1 \ldots M$. After the experiment ends, the simulator computes the entropy of the probability distributions defined by $\Pr[m_i = m_t]$ to evaluate anonymity, as:

$$-\sum_{i=1}^{M} \Pr[m_i = m_t] \cdot \log_2(\Pr[m_i = m_t])$$

B XRD

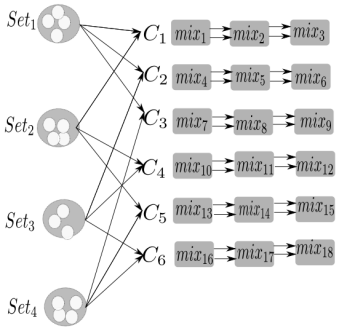

Figure 6: Arrangement of cascades in XRD.

Figure 6 represents the scenario where we simulate $C = 100$ clients, that results in 4 sets of 25 users each (as per XRD scheme). Every set is connected to one group of 3 cascades. Suppose we have 6 cascades C_i, $i = 1 \ldots 6$, then users' sets are connected to the following cascades respectively: $Set_1 = \{C_1, C_2, C_3\}$, $Set_2 = \{C_1, C_4, C_5\}$, $Set_3 = \{C_2, C_4, C_6\}$, $Set_4 = \{C_3, C_5, C_6\}$. It must be noted that each pair of user sets intersects with every other set in at least one cascade. This scheme guarantees that the set of users is not split into disjoint subsets.