Provable Anonymity for Networks of Mixes

Marek Klonowski and Mirosław Kutyłowski

Wrocław University of Technology, Poland

7th Information Hiding Workshop Barcelona
06.06.2005

Do we need anonymity?

Yes, we do:

- business to business communication
- privacy protection
- economic and political security of a country

how to hide information that two parties are communicating?

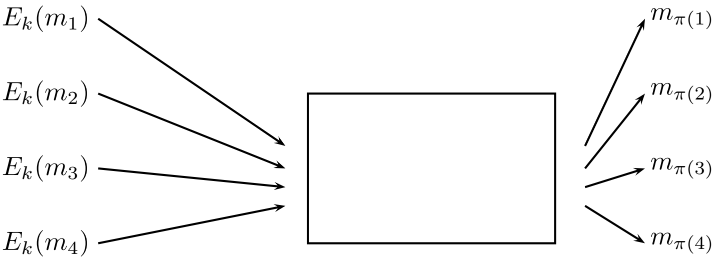
What is anonymity?

A.Pfitzmann, M.Köhntopp, 2000:
"Anonymity is the state of being not identifiable within a set of subjects, the anonymity set"

Measures of anonymity:

- cardinality of anonymity set
- consider probability distribution of possible destinations for a single input, entropy of this distribution as anonymity measure
- ...

Technical solution – a MIX



MIX -details

Parameter: k – public key of a MIX server and encryption scheme E

Processing: messages $m_1, m_2, m_3, \ldots, m_n$ to be published anonymously:

- the users submit $E_k(m_1), E_k(m_2), E_k(m_3) \ldots, E_k(m_n)$ to the MIX-server,
- the MIX-server
  - decrypts the ciphertexts,
  - chooses permutation π at random,
  - outputs $m_{\pi(1)}, m_{\pi(2)}, m_{\pi(3)} \ldots, m_{\pi(n)}$

Single mix solution

- as long as the mix is honest that mixing is perfect, but ...
- the mix knows everything and can betray this information,
- scalability problems.

Networks of mixes

Connect mixes into networks:

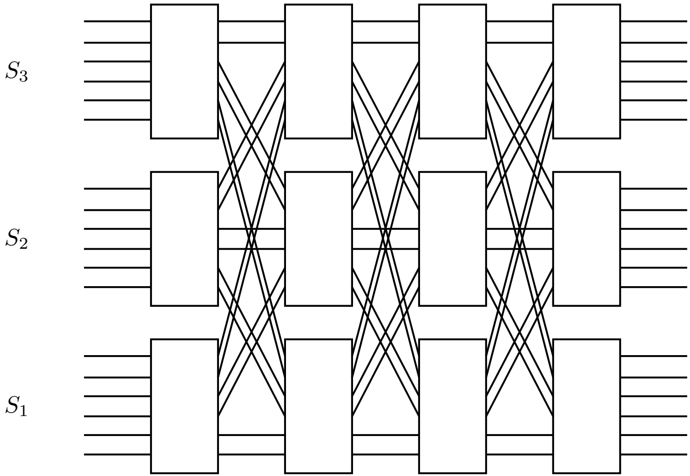
- the messages processed by many mixes in parallel, each mix responsible for a different group

Networks of mixes

Connect mixes into networks:

- the messages processed by many mixes in parallel, each mix responsible for a different group
- repeat after reassigning messages to groups

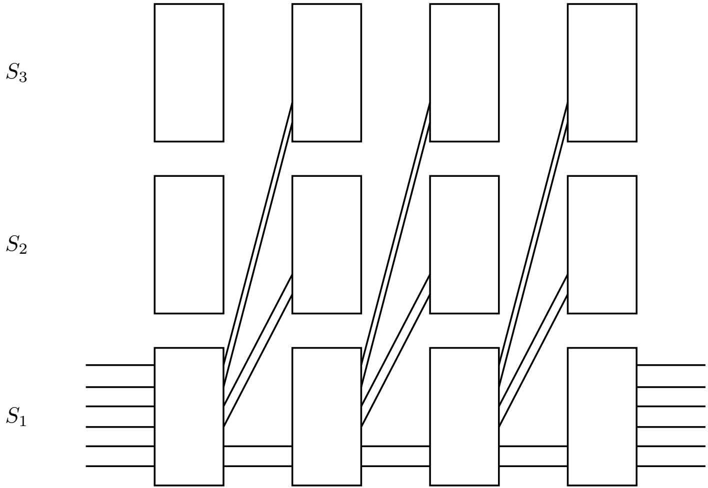
Parallel MIX Cascade



Parallel MIX Cascade

- k MIXes working in parallel,
- N messages,
- at each round a MIX processes N/k messages ...
- and splits the output into k groups of N/k^2 messages each, each group goes to a different MIX.

Parallel MIX Cascade



Security of PMC

- Is PMC mixing well the set of all messages?
- How many stages are required ?

Previous work

Philippe Golle, Ari Juels, “Parellel Mixing”, CCS’04

- A slightly different protocol
- Analysis of efficiency
- Anonymity definition does not take into account dependencies between messages.

More rigorous approach

- Each MIX chooses permutations uniformly at random.
- Do we get a random permutation of N elements
  so that each permutation has probability $\frac{1}{N!}$?

More rigorous approach

- Each MIX chooses permutations uniformly at random.
- Do we get a random permutation of N elements so that each permutation has probability $\frac{1}{N!}$?

Exact uniform distribution is never reached by PMC.

More rigorous approach

- Each MIX chooses permutations uniformly at random.
- Do we get a random permutation of N elements so that each permutation has probability $\frac{1}{N!}$?

Exact uniform distribution is never reached by PMC.

How close we get to it?

Definition based on Total Variation Distance

- output is a permutation of the input, described by a random variable Π_t
- quality of mixing defined as *total variation distance* between Π_t and the uniform distribution U:

$$TVD(\Pi_t, U) = \frac{1}{2}\sum_{\pi}\left|\Pr(\Pi_t = \pi) - \frac{1}{N!}\right| .$$

Definition based on Total Variation Distance

- output is a permutation of the input, described by a random variable Π_t
- quality of mixing defined as *total variation distance* between Π_t and the uniform distribution U:

$$TVD(\Pi_t, U) = \frac{1}{2} \sum_{\pi} \left| \Pr(\Pi_t = \pi) - \frac{1}{N!} \right| .$$

- Our goal is to estimate

$$\tau(\varepsilon) = \min\{T : \forall t \geq T \quad TVD(\Pi_t, U) \leq \varepsilon\} .$$

Main Result

Theorem

For a parallel MIX cascade

$$TVD(\Pi_t, U) < \frac{1}{N}$$

for $t > T$, where $T = O(\log k)$.

Remark

T does not depend on the number of messages N

Technical tools

- modeling as a Markov chain with a fixed initial state,
- estimating convergence rate to the uniform distribution - proving “rapid mixing” property

Technical tools

- modeling as a Markov chain with a fixed initial state,
- estimating convergence rate to the uniform distribution - proving “rapid mixing” property

- method used:
  - Delayed Path Coupling [A. Czumaj, M. Kutyłowski, 2001]
  - extension of Path Coupling [B. Bubley,M. Dyer, 1997]
  - other minor combinatorial and probabilistic techniques.

Coupling techniques

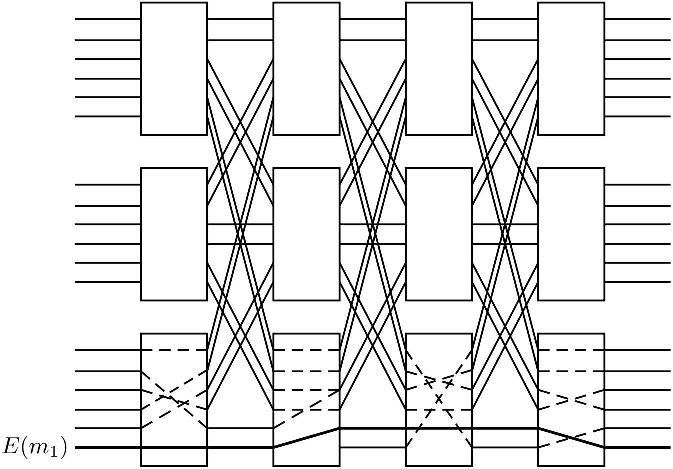
- Many variants
- TARGET:
  Estimate convergence rate of stochastic process Y_t
- Build two copies of process Y_t - say $(Y_t, Y_t^\star)$
- $Y_t^\star$ and Y_t have the same distributions but can be dependent.
- Convergence rate is related to the distance between the states of $Y_t^\star$ and Y_t.
- CORE OF THE PROBLEM: design dependencies so that the processes converge fast.

Technicalities

- It is enough to consider convergence for very special pairs of states.
- It is not necessary to define dependencies over one step - a group of steps may be considered.
- some combinatorics ...

Full proof in the paper.

Single Dishonest Server Case



Dishonest server case

- If at least one server is dishonest, then the number of required steps is $T = \Omega(\log n + \log k)$
- A single dishonest server really matters!

Remarks and open problems

- Is $\Omega(\log k)$ a lower bound?

Remarks and open problems

- Is $\Omega(\log k)$ a lower bound?
- The proof depends on the regular structure of the network. For random networks it should work as well.

Remarks and open problems

- Is $\Omega(\log k)$ a lower bound?
- The proof depends on the regular structure of the network. For random networks it should work as well.
- The proof should work also if each mix reveals its permutation used in a step with a certain probability (of course it influences the number of steps necessary).

Thanks for your attention!