Evaluation of Anonymity Providing Techniques using Queuing Theory

By Sarang Sharma, 04010135

This paper uses the ideas and principles of queuing theory to evaluate the effectiveness of methods proposed for anonymity. The paper describes the MIX method which is the most standard theoretical method for achieving anonymity although it is not practical in implementation. Hence, several variants of the MIX method have been proposed over the years by various researchers. The present paper attempts to evaluate the level of security achieved by each of the different methods by using queuing theory.

Owing to the successful work done over the years in cryptography messages can be securely transmitted and received but the use of cryptography alone cannot guarantee anonymity. An attacker can still observe the sender of a message and follow the message up to the receiver, thereby detecting the communication relation without the need to read the content of the packets. Hence, the decisive point of anonymity techniques is to organize additional traffic in order to confuse the attacker and conceal the particular communication relationship.

The MIX concept

The classical approach to implement anonymity is the MIX concept. In this concept an intermediate node called MIX is introduced. This node MIX collects packets from distinct users (anonymity set) and processes them so that no participant, except the MIX itself and the sender of the packet, can link an input message to an output message. This is achieved by following the following simple protocol:

User protocol: All generated data packets with address information are padded to equal length (agreement), combined with a secret random number RN, and encrypted with the **public key** of the MIX node.

MIX protocol: A MIX collects **n packets** (called batch) from **distinct senders** (identity verification), decrypts the packets with its **private key**, strips off the RNs, and outputs the packets in a different order (lexicographically sorted or randomly delayed) to the receiver.

The corresponding protocols described in the paper are applicable for the scenario where there is a sequence or cascade of the so called MIXs nodes which provide added security.

The reordering of the packets along with a change of externally observable bit pattern ensures the anonymity of the sender.

The (n-1) attack

The (n-1) attack refers to the attempt by an attacker to violate anonymity of a sender. If the anonymity set is of size is n and the attacker contributes (n-1) of members, then the remaining one is of course observable. Hence, the attacker is successful as long as it can fool the MIX that all the (n-1) packets which it sends are actually from different sources. In general the attacker is able to achieve this over the internet and can hence track the message from its source to destination.

If the attacker is sending messages at a constant rate λ_A and the rate at which real messages arrive is given by λ, then the probability that the attacker succeeds in tracking a particular real message is calculated by using basic queuing theory. It depends on the ratio of the two rates and the graph for different batch sizes looks like the following -:

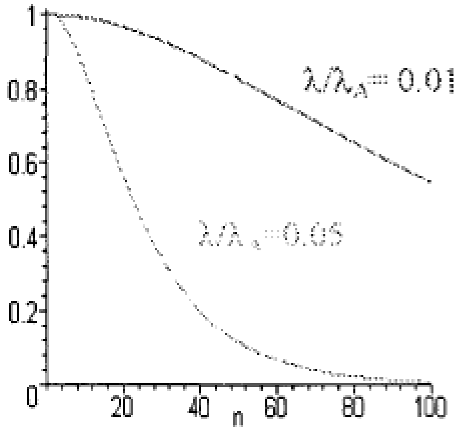

Figure 2: Probability of success for an attack on a message passing through a MIX with constant batch size n

The MIXmaster method:

In this method a batch is not used but a pool is used. The Mixmaster collects n messages in the pool mode and when the (n+1)th message arrives, the Mixmaster chooses one of the n+1 messages randomly and sends it forward. The probability of an (n-1) attack becoming successful drastically reduces in this case.

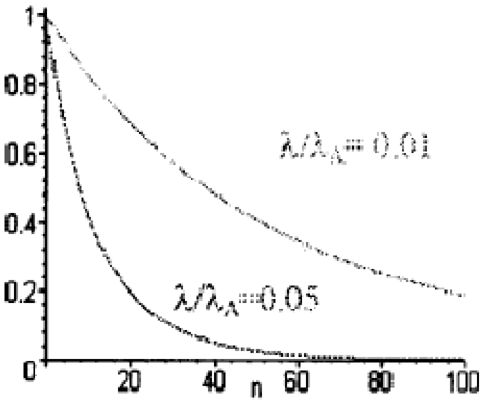

Figure 3: Probability of success for an attack on a message passing through a MIXmaster with constant pool size n

MIXes capable of Delaying

MIXes capable of delaying messages independent of traffic avoid the direct dependence on the packet arrival rate and are therefore not vulnerable to the attacks shown in the previous section. The paper has described two such protocols **T-MIX** and **SG-MIX**.

In T-MIX, instead of waiting for n messages the MIX waits for a **fixed time T** before sending out messages. Hence, messages from an active attacker do not influence the behavior of the T-MIX at all and therefore a disclosing attacker has no advantage over a passive eavesdropper. The only chance with the attacker is when only one message arrives in the decided interval T. This probability can be made arbitrarily small by varying T.

SG-MIX is even more secured than T-MIX and even does not require identity verification.. Here the sender selects with certain probabilities the delays to be made while transmitting the message forward. This set of delays is appended to the message and sent to the MIX nodes. Each of the MIX nodes waits for the specified amount of time. Hence, the eavesdropper can detect message flow only when there is only one message in the system .The security of the SG-MIX **does not rely on shuffling** a batch of messages but **rather on delaying** each message individually and independently by a random amount of time. As it is evident the probability of the success of an attacker is negligible with a reasonable choice of parameters. The exact expression for this probability is not derived in this paper.

Delay analysis

For delay analysis in transmission of a message through MIX nodes the author models the situation rightly as an M/D/1 system. He then using basic queuing theory concepts like the Little's Law to get expected delays. The following graph is hence obtained:

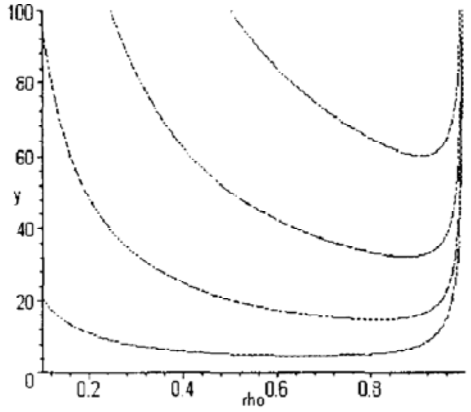

Figure 5: Mean total message delay $y=E(t_X)$ in a MIX with constant different batch sizes n (bottom-up) $n=5$, $n=20$ and $n=50$ and $(\mu_D=1)$

Conclusion:

To conclude I would say that in this work the authors have analyzed the MIX concept and some variants with the aid of queuing theory. We show that queuing theory is not only good for performance evaluation of these systems but can also be used for security evaluation. This paper enlightens us that in many applications queuing theory concepts may prove to be immensely useful. The identification of an appropriate queuing model for a system is not only interesting but also fruitful for predicting the behaviour of the system.