# Minstrels: Improving Communications Availability via Increased Relationship Anonymity

Ognjen Vukovic, György Dán, Gunnar Karlsson
School of Electrical Engineering
KTH, Royal Institute of Technology, Sweden
vukovic@ee.kth.se, gyuri@ee.kth.se, gk@ee.kth.se

## I. Introduction

Communication with high availability between a set of nodes on a pairwise basis is required in many systems, for example in modern industrial communication networks [1], [2]. Cryptography can provide authentication, confidentiality and data integrity but source and destination addresses are still visible to an outside attacker who observes one or more links. Hence the outside attacker can easily identify traffic patterns, who is communicating with whom, when and how often. Using this information the attacker may drop some important messages or perform bandwidth attacks (e.g. low rate TCP attacks) on the communication between two particular nodes. These attacks might be undetected and interpreted as high packet loss.

Relaying is one way to mitigate such attacks by hiding the sender and the receiver. With relaying the sender does not send the message directly to the receiver but via a sequence of other nodes. But even inside a closed group some nodes can be corrupted and behave as an attacker performing traffic analysis. The main goal of such attackers would be to determine the sender-receiver pair for messages that are relayed over them. For defence against the inside attackers a number of solutions were proposed (e.g. [3], [4]). For example in Crowds [4] anonymity is achieved through relaying. The sender of a message picks some node at random and sends a request for establishing a path. Information about the receiver is sent inside the request. The next node, the first relay, decides whether it should relay the request further (with probability $P_f$) or it should send it to the receiver (with probability $1-P_f$) where $P_f$ is a system parameter. This process is repeated by each relay node. The relaying path, once established, is used for some period of time. The major problem with Crowds is that the path length can be arbitrarily long, so it is not suitable for delay sensitive traffic. Since each relay node has the information about the receiver, only the sender identity is hidden from the inside attackers (sender anonymity). If the attacker can figure out the sender, it can identify a sender-receiver pair for the message, and may thereby violate relationship anonymity.

In this work we propose a system that improves Crowds in terms of relationship anonymity and in terms of the worst case path length.

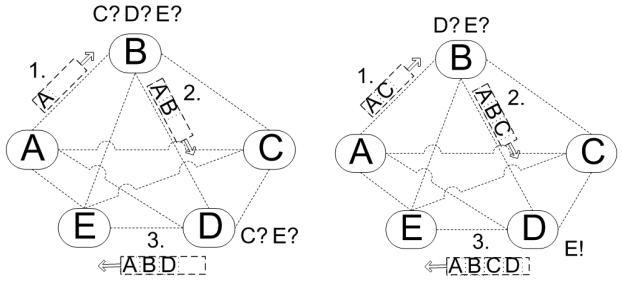


Fig. 1. The simple example of design

## II. System description

We consider a system with N nodes. When some node wants to send a message to another node it can forward the message to N-1 other nodes: directly to the receiver or to one of the remaining N-2 nodes for relaying. The decision is made uniformly at random among the N-1 options. When a node receives a message, it checks if it is the receiver by trying to decrypt the message, or a part of it. If the check fails, the message is sent further and the next hop is chosen uniform at random. Note that a relay node does not know who is the receiver, it can only check if it is the receiver itself.

To prevent very long paths, where some relay nodes may be visited multiple times, the messages record a list of the nodes already visited. When a relaying node receives a message, it can relay the message only to nonvisited nodes. To decrease the path length the sender can initialize the list of visited nodes with some of the relaying nodes. These initialized nodes are considered as visited so that message can not be relayed to them. The sender initializes the list with $M \in \{0, ..., N-1\}$ nodes with probability $P_M$, where $\sum_{M=0}^{N-1} P_M = 1$. For M=0 the list is empty, for M=1 the list is initialized only with the sender and for $M > 1$ the list is initialized with the sender and M-1 other nodes. The list must not be initialized with the receiver, because the message would not reach it. The distribution of $P_M$ is a system parameter.

Figure 1 shows a simple example with five nodes. When node A wants to send a message to node E it can initialize a list empty, with itself or with one up to three relaying nodes (B,C,D). Figure 1a shows a case where the list is initialized with sender A and is sent to B. Node B determines that it is not the receiver, puts itself in the list and chooses uniformly at random the next relay among the remaining nodes (C,D,E).

The next hop, node D, follows the same procedure with only two forwarding options (C,E). When the message eventually reaches node E, node E will determine that it is the receiver. In Figure 1b we can see another example where the list is initialized with the sender and node C, and the message is sent to B. Node B determines that it is not the receiver, adds itself to the list and decides to which of the remaining nodes (D,E) to send the message. Node C is considered as already visited.

## III. Performance Evaluation

To evaluate our solution we use two metrics: expected path length and relationship anonymity. The expected path length is the average number of nodes on the path including the sender and the receiver. It depends on the number of nodes (N) and on the initialization of the list of visited nodes. For brevity we show the expression for a simple scenario: the sender always initializes the list of visited nodes with itself ($P_1 = 1$). For a system with N nodes the path length K is uniformly distributed on {2, ..., N} and the average path length is:

$$E[K] = \sum_{k=2}^{N} k \times P(K = k) = \sum_{k=2}^{N} k \frac{1}{N-1} = ... = \frac{N+2}{2}.$$

The other metric we use is the relationship anonymity. It depends on the probability of having an attacker on the path and on the probabilities that the attacker assigns to the sender (that it sent a message) and to the receiver (that it is the final destination) when it gets a message. The probability that an attacker can assign to the sender depends on the list of visited nodes and on the node that the message is received from. The probability that an attacker assigns to the receiver is easily calculated from the list as $1/H$, where H is the number of nonvisited nodes that are not attackers. The sooner the attacker gets the message the higher probability it can assign to the sender and the later the message is caught the higher probability the attacker can assign to the receiver. In general the relationship anonymity can be expressed as

$$P_{rel} = \sum_{i=2}^{N-C-1} P_{firstC}(i) P_{src}(i) \sum_{j=i+1}^{N-1} P_{lastC}(j) P_{dst}(j),$$

where C is the number of inside attackers, *i* is the position of the first attacker with probability $P_{firstC}(i)$, $P_{send}(i)$ is the probability that the attacker can assign to the sender, *j* is the position of the last attacker with probability $P_{lastC}(j)$ and $P_{rec}(j)$ is the probability that the attacker can assign to the receiver.

For brevity we show results for the simple case when the sender either leaves the list of visited nodes empty or it puts itself only ($P_0 + P_1 = 1$). We also show results for Crowds for different values of $P_f$.

Figure 2 shows the expected path length E(K) for Crowds and Minstrels as a function of $P_f$ and $P_1$ respectively. The path length for Crowds is almost geometrically distributed with mean $2 + \frac{P_f}{1-P_f}$ [4], and it increases fast for large $P_f$. Minstrels shows a slight decrease of the expected path length as $P_1$ increases. Initialization of the list with more nodes would decrease the expected path length.

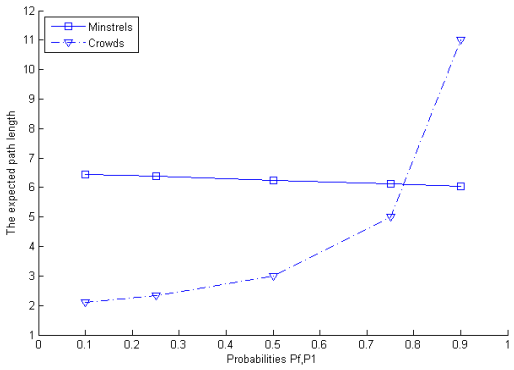


Fig. 2. The expected path length (E(K)) vs. $P_f, P_1$ for N=10

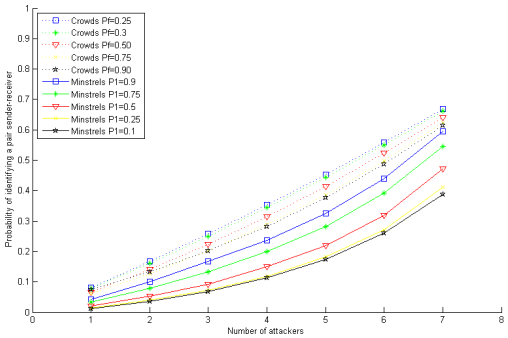


Fig. 3. The relationship anonymity ($P_{rel}$) vs. number of attackers for N=10

Figure 3 shows the relationship anonymity $P_{rel}$ for Crowds and Minstrels for various number of attackers C. As the number of attackers increases, the probability of identifying a sender-receiver pair increases which means that relationship anonymity degrades. For all values of $P_1$ Minstrels achieves better relationship anonymity than Crowds because it attempts to hide both the sender and the receiver of the messages.

## IV. Conclusions and Future Work

In this work we presented a system that improves Crowds in terms of relationship anonymity while at the same time it provides bounded path length. For brevity we presented results for a simple case which already showed some improvements compared to Crowds. This work is the first step towards providing communication with high availability on a pairwise basis for a group of nodes and we intend to extend our solution with hiding traffic patterns and with load balancing.

## References

[1] Dzung D., Naedele M., Von Hoff T.P., Crevatin, M. Security for Industrial Communication Systems. *Proceedings of the IEEE Volume: 93 , Issue: 6*, pp. 1152 - 1177, 2005.

[2] Chee-Wooi Ten, Chen-Ching Liu and Govindarasu Manimaran Vulnerability Assessment of Cybersecurity for SCADA Systems. *IEEE Transactions on Power Systems, vol. 23, no. 4*, 2008.

[3] Paul Syverson, D. M. Goldschlag, and M. G. Reed. Anonymous connections and onion routing. *In Proceedings of the IEEE Symposium on Security and Privacy*, pages 4454, Oakland, California, May 1997.

[4] Michael K. Reiter and Aviel D. Rubin. Crowds: Anonymity for Web Transactions. *ACM Transactions on Information and System Security*, pp. 66-92, 1998.