The Tale of Discovering a Side Channel in Secure Message Transmission Systems

Majid Ghaderi[1], Samuel Jero[2], Cristina Nita-Rotaru[3],
Hamed Okhravi[2], and Reihaneh Safavi-Naini[1]

[1] University of Calgary
[2] MIT Lincoln Laboratory
[3] Northeastern University

Abstract. Secure message transmission (SMT) is a cryptographic system that provides information theoretic security and reliability for point-to-point message transmission in a network that is partially accessible to the adversary. This is the story of a research project that started with a research question about the cost of security for an SMT system with theoretical security analysis and, unexpectedly, ended with discovering a side-channel that affects the security of implementations of a wide range of SMT systems.

1 Introduction

Cryptography needs assumptions about the adversary to achieve security. Computationally secure systems use assumptions on the computational power of the adversary. A second type of assumptions, pioneered by Aaron Wyner [6] in the study of wiretap model, is called physical layer assumption and assumes the adversary has limited access to the physical environment. Information theoretic secure systems use such assumptions about the network access of the adversary.

Dolev, Dwork, Waarts, and Yung [1] modelled the network between a sender and a receiver as *a set of node-disjoint paths*, and showed that, assuming that the adversary's has only access to a (proper) subset of all paths, one can achieve perfectly secure and reliable communication. The adversary can eavesdrop on a path, block the communication, or be a Byzantine adversary and change the message. Security is achievable if the number of paths that can be accessed by the adversary is below a threshold that depends on the type of the adversary. Security of the system relies only on the physical assumption of accessibility of paths to the adversary, and no limit is assumed on the adversary's computational power.

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited. This material is based upon work supported by the Under Secretary of Defense for Research and Engineering under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Under Secretary of Defense for Research and Engineering.

1.1 SMT with Passive Adversary

If there are n node-disjoint paths between the sender and receiver, security can be achieved if up to $n-1$ paths are eavesdropped. Security is achieved by using a simple randomized message encoding algorithm for a message m where an (n,n) secret sharing scheme generates n shares, and sends each share on one of the paths. If one can assume the adversary can access at most k, $k<n$ of the n paths, one can use a random k-subset of the paths and use (k,k) secret sharing. This will improve information rate of the communication from 1/n to 1/k. Although the set of k path is randomly chosen and not known to the adversary, if they stay fixed for transmission of many packets that form the message, the adversary who can only access a limited number of paths, will be able to probe paths one at a time and learn all the target paths over time, in which case security will be lost.

1.2 Moving Target Defence (MTD)

Moving Target Defence (MTD) systems [3] use randomization and dynamism in the system to achieve security. By randomly moving the attacker target in consecutive time intervals, the attacker's window of opportunity is reduced, and because of the randomization their effort in one time interval will (partially) lose its value in the next time interval. Safavi-Naini et al [5] considered a basic MTD system to achieve post-quantum (information theoretic) security in the following setting. Alice and Bob are connected with a set of n node-disjoint paths, k of which can be eavesdropped by the adversary. Time is divided into time-slots. In each time slot Alice randomly chooses a subset of k target paths to send the shares of a (k,k) shared message over the paths (as described above). The eavesdropper also chooses a subset of paths to eavesdrop and as long as not all the target paths are known to the adversary, the transmission will be secure.

Analysis of the system uses a Markov chain with $k+1$ states, where states are labeled by $i \in \{0,1,\cdots,k\}$, and state i corresponds to the adversary's knowledge of i target paths. The Markov chain models a game between the *defenders* (Alice and Bob), and the adversary. The winning state is the state k where the adversary knows all the target paths. The analysis assumed defenders play a basic *memoryless strategy* and in each time slot, with a probability λ decide to move, in which case they randomly choose one of the k current target paths, and re-allocate it to a randomly chosen path from the remaining $n-k$ paths. The adversary also moves with probability μ in each time interval, and if they do, they randomly select one of the paths from those that they have not learned yet. The probability μ depends on λ and a second parameter τ that is the probability of being detected because of the move.

Security. The two security measures that are considered are, (i) *the expected number of times that the adversary wins in the first T time slots,* given by $T.\pi(k)$ where $\pi(k)$ is the Markov chain stationary probability of state k, and (ii) *Expected number of steps until the first compromise happens*, denoted by $E^{(1)}_{win}$. These measures can be computed using transition probability matrix of the Markov chain, which can be obtained using strategies of the defenders and the attacker. Graphing the expressions of these results for typical settings suggests that increasing λ results in an increase in $E^{(1)}_{win}$, and thus higher security. This matches the intuition that faster changes in the system, will "increase" security in the sense that it will take longer for the attacker to find a message.

2 Understanding the Cost of an MTD System in Real Networks

Implementing dynamism in a real-world network is costly: hopping a path requires sufficient time so that the packets on the existing paths can reach their destination, and switches on the new path can be activated and updated for the new path. This suggests that, depending on the network topology, there will be a bound on λ beyond which, the network transmission would not be reliable in practice, in particular because all shares are required for the reconstruction of the message. This motivated the need for implementation and experiments to get a handle on this cost and determining meaningful values of λ for a given network. This is what we set out to do. Our research question was: *What is the cost of hopping paths in networks, and how the hopping rate can be determined?.*

2.1 A Software Defined Networking (SDN)Approach

We chose SDN to implement and study our research question. SDN is a paradigm that separates the network into a control plane, concerned with determining how the network should forward data, and a data plane that actually forwards the data. The data plane consists of programmable switches able to match on a variety of packet fields and perform basic forwarding actions while the control plane is implemented through a logically centralized SDN controller that coordinates and controls these switches. The SDN controller provides a framework for defining, enacting, and enforcing per-flow policies in a dynamic manner. In SDN, a controller switch determines the paths that a packet takes in the network, and sends control messages to switches in the network to set up those paths.

The first set of experiments (2018 and 2019) used Minitnet network emulation platform, to implement the path-hopping scheme over SDN. The emulation, however, indicated the need for a real testbed implementation in order to accuratly measure the hopping cost. In the subsequent six months, we purchased 2 multiport SDN switches, learned how to configure and program them, and used them to set up a small 20 node network to estimate the cost of path hoping. The experiments again could not capture the cost. Modern switches could perform the required path switching in negligible time for the small test network that we had set up.

We decided to refocus the project and explore other aspects of the real-world implementation of the system. We believed the approach had very attractive properties and if it could be securely and efficiently implemented, it can provide solution to post-quantum secure communication in some real-world settings (e.g., establishing initial communication when no other pre-shared keys are available). The important advantages of the system were: (i) information theoretic security, resilient against an adversary with access to a quantum computer, (ii) no requirement for shared keys, (iii) no reliance on computational assumption, and (iv) future proofing, in the sense that an adversary who stores their available transcript of the message, will not have any advantage in analyzing the stored transcript in the future.

We considered the following design goals for the implementation. 1) Ensuring that shares are carried over disjoint paths, and not merely assume that this is the case. 2) Being able to handle real network behavior. This includes the packet losses that are inevitable in real networks and the fact that multiple connections between different or the same endpoints may be in progress at any time. 3) Being implementable on standard SDN hardware. We wanted a protocol that would work with hardware that one can buy today. In the following few months, we implemented an application for transporting arbitrary messages using path-hopping, and reported our experiments on the effect of hopping interval on packet loss in a conference paper. We submitted the paper to a leading conference in networking and system security[4].

The paper was rejected. The main criticism was, *how can one estimate k, the number of paths that the adversary can hold at a time.*

3 Discovering a Side-channel

Until this point, our goal was to show feasibility of using path-hoping in practice. We have had extensive experiments on Mininet and a physical testbed. Analyzing this data, however, showed a fundamental security problem, that although in retrospect, was obvious, was neither part of our initial research goal, nor part of the implementation and experiments that we had planned for showing feasibility of path hopping in practice.

Our experiments showed that in transmitting shares of a message, the receiver node must wait to receive all the shares, and this waiting time depends on the length of the longest path that is used for carrying shares. The shares stay in the network for different lengths of time, and this gives the attacker the opportunity to access the shares on the longer paths of the previous time intervals, thus effectively finding all message shares of a previous interval. The attack worked as long as the system used fixed-length time interval, and the attacker was limited to accessing up to k path at a time. The attack would not have worked if the next transmission occurs after acknowledgement for the receipt of a sent message. However, this would be a costly process and makes the system unusable in practice. This was an important discovery. Our experiments had unraveled a side channel that exists in implementation of *all* multi-path schemes that model the network as a set of n paths, and limit the adversary to only having access to up to k paths. That is, the attack goes well beyond path-hopping systems and shows the need for a refined SMT model when used in real networks.

3.1 Network Data Remanence Side-channel

We called the side-channel, Network Data Remanence (NDR) side-channel. This was inspired by data remanence side-channels that are defined in the NSA/NCSC Rainbow Series as “the residual physical representation of data that has been in some way erased” [2]. We observed that modeling the network as a set of n paths and assuming the attacker can access up to k paths effectively means that *the attacker has a single chance to capture*

[4] The paper was entitled *Software Defined Network-based Path Hopping for Quantum-safe Secure Communication.*

a packet (a.k.a. a share) on a path. In real networks, a path consists of multiple hop, each with an associated delay. Thus, shares linger in the network, creating a side-channel that can be potentially exploited by an attacker to break the perfect secrecy guarantee of the secret-sharing schemes. Our new research goal became studying this side-channel, and proposing mitigation mechanisms for it. We followed this new direction in the subsequent months. The new submission was accepted and presented at NDSS 2021 [4].

4 Reflections

The project took over three years and became a true exploratory work: we set out with one research goal and ended up with a surprising discovery that completely changed the direction of the project.

It showed us once again the importance of implementation and experiments in using theoretical results in practice. Implementing cryptographic systems that rely on physical assumptions are significantly more challenging than the implementation of computationally secure system, and critically relies on collaboration across disciplines.

Our implementation underlined the importance of vetting assumptions in real-world settings. In the network model that we considered, one needs to at least guarantee (i) paths are truly disjoint (e.g., not implemented through overlay networks that share nodes and links in lower network layer), (ii) the estimate of the adversary's power is "verifiable" (e.g., by supporting evidences). Protection against our NDR side-channel requires further provisions (e.g., encoding proposed in [4]).

We learned how to collaborate across disciplines, cryptography, system security, and networking. The same research question finds different statement, different evaluation criteria, and approaches to evaluation. In physical security, this collaboration is essential as security is tightly related to the lower-level network properties.

References

1. Danny Dolev, Cynthia Dwork, Orli Waarts, and Moti Yung. Perfectly secure message transmission. *Journal of the ACM (JACM)*, 40(1):17–47, 1993.
2. NSA NCSC. Covert channel analysis of trusted systems (light pink book). *NSA/NCSC-Rainbow Series publications*, 1993.
3. H. Okhravi, T. Hobson, D. Bigelow, and W. Streilein. Finding focus in the blur of moving-target techniques. *Security Privacy, IEEE*, 12(2):16–26, Mar 2014.
4. Leila Rashidi, Daniel Kostecki, Alexander James, Anthony Peterson, Majid Ghaderi, Samuel Jero, Cristina Nita-Rotaru, Hamed Okhravi, and Reihaneh Safavi-Naini. More than a fair share: Network data remanence attacks against secret sharing-based schemes. In *28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, February 21-25, 2021.* The Internet Society, 2021.
5. Reihaneh Safavi-Naini, Alireza Poostindouz, and Viliam Lisy. Path hopping: An mtd strategy for quantum-safe communication. In *ACM Workshop on Moving Target Defense*, pages 111–114, 2017.
6. A.D. Wyner. The wire-tap channel. *Bell Systems Technical J.*, 1975.