

Together We are Strong: Securing Industrial Wireless Networks with Overhearing Nodes

Sebastian Henningsen

Lucas Mann

Stefan Dietzel

Björn Scheuermann

Humboldt-Universität zu Berlin
 Berlin, Germany
 {sebastian.henningsen, stefan.dietzel}@hu-berlin.de
 {mannluca, scheuermann}@informatik.hu-berlin.de

Abstract—Recent works have shown that the wireless link between two nodes is neither a shared secret nor unforgeable by an adversary – questioning the effectiveness of many physical layer security mechanisms. In this work, we argue that physical layer security is still valuable when multiple nodes combine their measurements of overheard adversarial transmissions. This assumption is applicable to, e.g., industrial wireless networks, where nodes are spatially close and wireless links are unique. First, we design a novel physical layer adversary detection algorithm particularly suited for the scenario of multiple sensing nodes. Second, we evaluate our algorithm with real-world measurements, which show that already four detecting nodes are sufficient to significantly increase the detection performance.

I. INTRODUCTION

Physical layer security leverages inherent characteristics of the physical medium, e.g., wireless channel properties, to detect adversarial anomalies. Usually, coarse-grained Received Signal Strength Indicator (RSSI) [1]–[3] or fine-grained Channel State Information (CSI) [4]–[6] measurements, so called *link signatures* are used.

These mechanisms assume that link signatures are an unforgeable and shared secret. In the recent past, however, these assumptions have been challenged by a number of attacks [7]–[9]. In particular, by choosing an appropriate input signal, an adversary is able to inject frames in such a way that the receiver measures a particular link signature [9]. Therefore, an adversary can inject packets with a forged link signatures of his choice (using software defined radios). As a consequence, link signatures cannot be deemed secret, nor are they unforgeable. As a remedy, Fang *et al.* [9] propose a countermeasure that operates on the PHY-layer, whereas our approach functions on the link layer.

We propose to combine multiple link signature measurements from sensing nodes to restrict the adversary's degrees of freedom. When forging link signatures for a single node, the attacker has *one* degree of freedom (the signal) and *one* constraint to match (the link signature measurement). If, however, multiple nodes overhear the same adversarial transmission and subsequently combine their measurements, the attacker still has *one* degree of freedom but *multiple* constraints to match. Effectively, the adversary faces an overdetermined set of equations and is, therefore, unable to simultaneously forge the link signatures at all overhearing nodes. The idea of

attacker detection with multiple sensors in the presence of an adversary has been raised in [10]. In contrast to their work, we focus on CSI measurements instead of RSSI.

The assumption of multiple nodes overhearing a single transmission holds, for instance, in the domain of industrial wireless networks, especially in real-time systems where nodes are often spatially close. Hence, even with directed antennas, an adversary is likely to encounter at least some nodes that overhear the adversarial transmission.

In this paper, we propose a new attacker detection algorithm, the *cluster count and classification procedure* (C^3P_o). C^3P_o is specifically designed to leverage the properties of scenarios with multiple sensing nodes. Our algorithm statistically analyzes link signature measurements, similar to related works such as CSITE [5] and [4]. A significant difference to our work lies in the setting: CSITE, for example, assumes a baseline of trusted, encrypted 802.11 data frames alongside potentially insecure management frames, whereas we make no such assumption. We evaluate the algorithm with real-world measurements conducted on commodity 802.11n boards [11]. Our measurements show that already few nodes are sufficient to significantly increase the detection performance in comparison to single-detector scenarios.

II. NETWORK MODEL

Our network model is motivated by low latency, closed-loop industrial wireless networks with real-time requirements [12], which are often organized in star topologies with a central master and several slave nodes [13]. C^3P_o is based on link signatures measured as CSI, which describe how the channel has altered a signal. Let a denote the original signal sent by the transmitter; then the signal at the receiver can be defined as follows: $b = ha + \eta$ (1), where η is random Gaussian noise and h is the complex channel impulse response (i.e., the link signature), which is estimated at the receiver using a known input signal a . A link signature h can be spoofed by choosing the input a_m such that the estimation of h at the receiver based on b_m results in the desired channel estimation h_m [9]. We assume a Byzantine [14] / Dolev-Yao [15] adversary with access to secret key material and arbitrary link signatures who seeks to violate authenticity and availability.

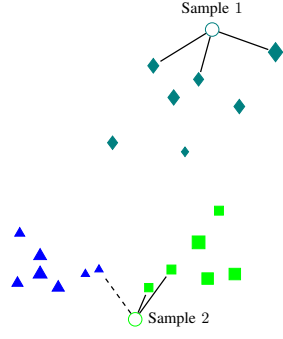


Figure 1. Examples of classification for two points. Sample 1 will be classified as legitimate, whereas sample 2 will be classified as an attacker-injected packet.

III. C^3P_o ADVERSARY DETECTION

We built an algorithm that translates CSI inputs into compact and exchangeable *decisions* on the attack probability for each received packet. The C^3P_o combines two main concepts to achieve adversary detection. A cluster-based detection component determines the initial probability of a new sample stemming from its claimed sender. This classification component is complemented with a cluster counting mechanism, which improves detection robustness despite changing link signatures and accumulations of adversarial link signatures. The classification component checks whether a new link signature fits into the cluster of the corresponding claimed sender. Using only the classification component, it could happen that adversarial link signatures form their own cluster, which would then lead to a small mean distance, leading to a wrongly high decision value. We can detect these separate clusters by estimating the total expected size of both individual sub-clusters, i.e., the cluster encompassing *all* data points of one claimed sender. If the actual size of the cluster is larger than its expectation, this is a strong indicator for an attack.

A. Information representation

As information base, the algorithm uses a measurement space as shown in Fig. 1 for two-dimensional example measurements; the actual measurement space is 30-dimensional. The algorithm uses past CSI samples for classification; these are illustrated by the triangle-, square-, and diamond-shaped points. The shapes correspond to three different senders whose CSI values are stored by the algorithm. In the example shown in Fig. 1, the senders form three clusters with different variances. A sample consists of a CSI value, which is represented by its location in the plane, as well as the packet's claimed sender, which is indicated by the symbol (and color). Additionally, a sample's decision value is indicated by its symbol's size. A high decision value (represented by large symbols) indicates trust in the claimed sender being the same as the actual sender. A low decision value of a CSI sample vector $\vec{c}$ indicates that the classification algorithm suspects that the packet was spoofed, i.e., the actual sender differs from the claimed sender. Defining decisions values in this way

Algorithm 1 Classification and Cluster Count Procedure

```

1: procedure  $C^3P_o(\vec{c}, E)$ 
2:    $\vec{d} \leftarrow$  euclidean distances to  $k$  nearest neighbors( $\vec{c}$ )
3:    $\vec{e} \leftarrow$  claimed senders of  $k$  nearest neighbors( $\vec{c}$ )
4:    $t_E \leftarrow$  current threshold for claimed sender
5:    $\vec{b} \leftarrow$  decision values of  $k$  neighboring samples

6:    $x \leftarrow \frac{1}{k} \sum_{1 \leq j \leq k: e_j = E} b_j \exp(t_E - d_j)$ 

7:   for sample  $\vec{s}_i$  in the most recent  $k$  samples do
8:     Check if  $\|\vec{s}_i - \vec{c}\| > \sqrt{\frac{m_e}{k}} \cdot t_E$ 
9:   if no sample  $\vec{s}_i$  violated the check then
10:     $x \leftarrow x + b_d$ 
11:  else
12:     $x \leftarrow \frac{x}{2}$ 

13:  ADDSAMPLE( $\vec{c}, E, x$ )
14:   $t_E \leftarrow \beta t_E + (1 - \beta) \text{mean}(\vec{d})$ 

15:  RETURN( $x$ )

```

allows nodes in the network to generate a global perspective by aggregating local decisions.

As samples from honest nodes can be acquired within a few microseconds in real-time industrial wireless networks, we assume that C^3P_o has saved link signatures of every benign sender within a very short training phase before encountering the first adversary.

B. Classification of new samples

In order to classify newly received samples, the algorithm takes as input a link signature in the form of a CSI sample $\vec{c} = (c_1, \dots, c_n)$, as well as the claimed sender E of a received data packet. The output is a decision value x , which is calculated using Algorithm 1.

Assume we want to classify the benign sample 1 in Fig. 1, which claims the diamond sender. In the first step of Algorithm 1, the euclidean distances to the k nearest neighboring samples are computed and stored in vector $\vec{d}$. This is illustrated in Fig. 1 by the three solid black lines (hence, $k = 3$) originating from sample 1. In the next step, the claimed senders of the k nearest neighbors are stored in the vector $\vec{e}$. In the example, $\vec{e}$ contains k times the diamond sender.

For each claimed sender E , a soft threshold t_E (cf. line 4) captures the expected dispersion of future CSI-samples of that particular sender in the measurement space. As shown in line 14, the threshold for a claimed sender is determined by the moving average over the mean distances of previously received samples by the same claimed sender. As link signatures for each honest sender may vary to different degrees, t_E is maintained separately for each claimed sender.

Next, in line 6, the initial decision value associated with sample 1 is calculated. The initial decision is determined by the decision values b_i of those neighboring nodes in

$\vec{e} = (e_1, \dots, e_k)$ that have the same claimed sender E as the new sample and the distance d_j to these neighbors compared to the expected threshold t_E . By incorporating b_i , adversarial packets can be detected although they may form their own cluster in the measurement space, thereby improving over existing algorithms [4]–[6].

We normalize the calculated sum using a factor of $1/k$. In doing so, we introduce an additional penalty for likely adversarial samples. Thereby, the diversity in $\vec{e}$, which may indicate adversarial activity, further lowers the calculated initial decision value.

This concept is illustrated by sample 2 Fig. 1: whereas sample 1's k nearest neighbors are all from the diamond sender, sample 2 has two square neighboring samples and one triangle sample. Sample 2 is supposed to come from the square sender, hence, the triangle sample does not contribute to the decision value calculation in line 6 (indicated by the dashed line). Since the neighboring samples of 2 already have a small decision value (as indicated by their size), the overall decision on 2 will be low.

C. Cluster count detection

As outlined before, a simple comparison of a sample's mean distance to its k nearest neighboring samples may not be sufficient in scenarios with rapidly changing link signatures. The adversarial samples may form a cluster, which cannot be detected by this calculation alone. Therefore, line 7 checks whether the samples of sender E form a single cluster or whether there are two cluster for the same sender E : a benign and an adversarial one.

The cluster counting is not only necessary to detect adversarial clusters, but it also helps to adapt to sudden changes in the channel. It only needs a few (approximately, k) samples to adapt to new channel conditions, since it compares a new sample to the k most recent ones rather than the – potentially older – k nearest ones. In contrast, the classification based on decision values takes a longer time to react to channel changes, thereby falsely classifying genuine packets as adversarial in the meantime.

For the cluster assessment, the expected cluster radius r_E is estimated using the moving average t_E . If a sample's distances to *at least one* of the k most recent samples of the same claimed sender E is larger than the cluster radius r_E , we deduce that there exists more than a single cluster for that sender. To minimize computational overhead, we approximate the expected size of the cluster r_E using the same threshold t_E that we use for the initial decision calculation. Assume there are m_E CSI samples of a sender E stored by the algorithm. These m_E samples will be uniformly distributed in a disc with an unknown radius r_E . The k nearest neighbors also form a disc with radius t_E . Due to the uniformity, the ratio of points in each cluster disc roughly equals the area ratio of the disc's areas:

$$\frac{k}{m_E} \approx \frac{\pi(t_E)^2}{\pi(r_E)^2} \Rightarrow r_E \approx \sqrt{\frac{m_E}{k}} t_E,$$

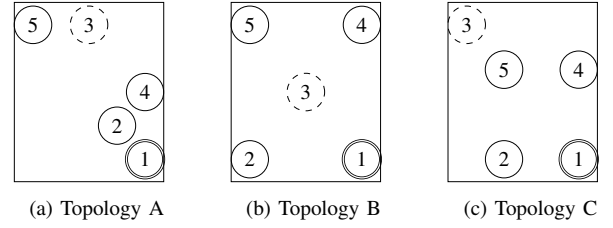


Figure 2. The three different measurement topologies. Node 1 is the master and node 3 serves as the attacker.

which yields a simple yet effective heuristic to approximate r_E for use in the cluster counting component. Finally, the newly classified sample is added to the measurement space, the exponential moving average t_E for the claimed sender E is updated, and the final decision x is returned. At this point, the new sample can be accepted locally if the decision x is larger than a system parameter b_a or rejected, otherwise.

IV. MEASUREMENT-BASED EVALUATION

In order to verify our hypothesis that cooperating nodes increase the detection performance, we evaluate an upper bound on the achievable performance of C^3P_o in the following. To this end, we aggregate multiple local measurements in order to form a global decision. Each node locally computes x_i which are then sent to and aggregated at the master node by taking the average. We refrain from assuming a specific protocol for sending local decisions to the master node [10], [14], [16], since it may distort the aggregation we wish to study.

The comparison to CSITE [5] and other related works is difficult due to different attacker models. As an architecturally similar approximation, we compare C^3P_o against a version of itself without the cluster counting aspect, “No-CC” (for no cluster counting). For a meaningful comparison we used the attacker model of [5]: the adversary injects spoofed frames with node 2's address into the system but does not spoof CSI values.

Three different topologies with 5 nodes, each equipped with Intel 5300 cards that allow to obtain CSI values, were set up as depicted in Fig. 2. The distances between nodes vary between ≈ 1 m and ≈ 5 m. Node 1 is the master node, whereas node 3 is the attacker. At each topology, 10 trials with ≈ 11400 sent frames (and thus CSI measurements per node) were conducted in a TDMA fashion: every node had a fixed time slot of 0.1 s for transmitting random data. We chose $k = 5$ (cf. [5]) and $\beta = 0.05$. All figures show the mean of all trials alongside with error bars corresponding to 0.95 confidence intervals.

In Fig. 3, we see the global sensitivity for $N = \{2, 3, 4\}$ sensing nodes. The topology is on the x -axis and the plot is divided to show the results for different choices of N . The results show that the detection performance of No-CC increases with the number of nodes and that just 4 nodes are sufficient to achieve almost perfect sensitivity. The global specificity is shown in Fig. 4, where the x -axis shows each

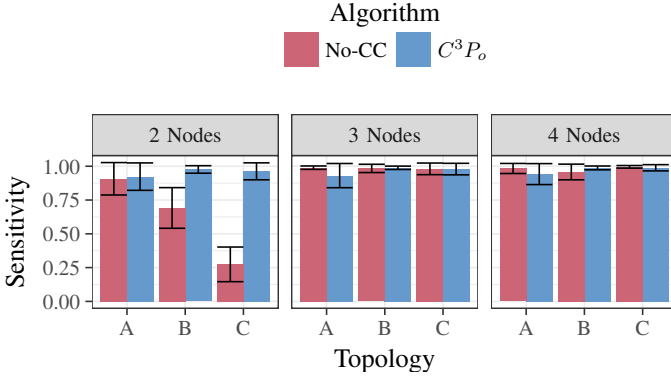


Figure 3. Global sensitivity (intrusion detection) of all receivers at different topologies and for $N = 2, 3, 4$. The aggregation of decision values increases the detection of adversarial frames, especially for the No-CC classifier.

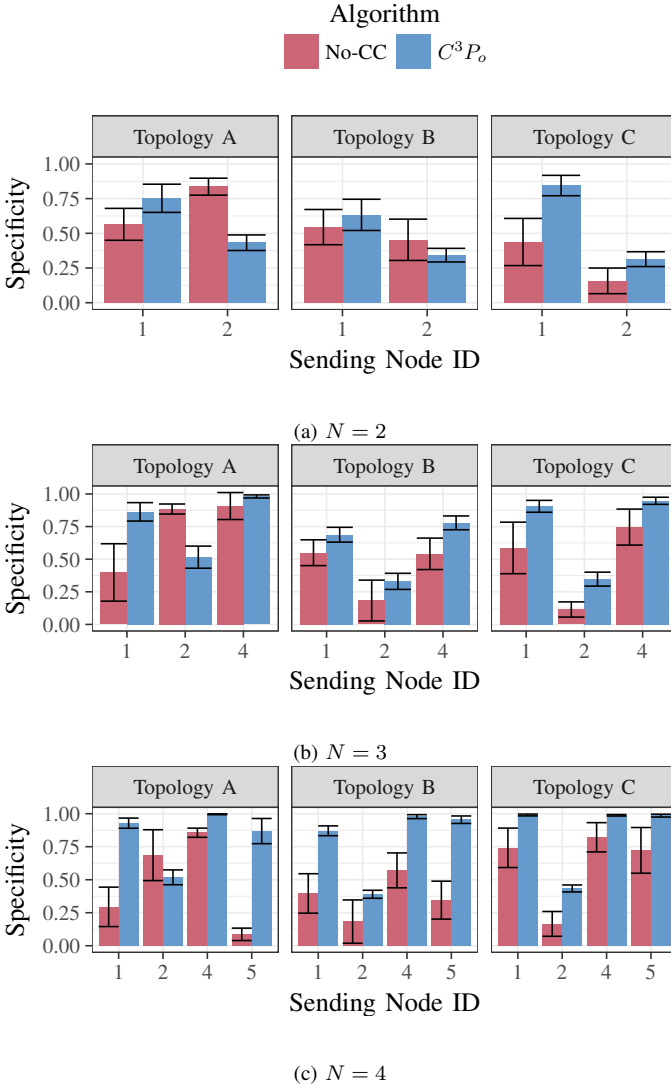


Figure 4. Global specificity (honest sender acceptance) rate at different topologies for $N = 2, 3, 4$. The aggregation of decision values increases the average specificity of C^3P_o by 49.1 percent.

sender and the division is by topology. The bar of sending node ID 1 for No-CC in Fig. 4a in topology A means that globally, approximately 50 percent of all frames sent by node 1 where classified correctly.

The topology has a significant impact on the results, but at least for C^3P_o the aggregation of decisions increases the specificity of C^3P_o by 49.1 percent between $N = 2$ and $N = 4$ honest sensing nodes. The decreased performance of correctly classifying frames from node 2 of C^3P_o is due to the attack: the adversary spoofs packets with node 2's address, leading to a constant distrust in this node. In summary, although the results vary between the different topologies, Fig. 4 clearly shows the benefits of combining multiple link-signature measurements when using C^3P_o .

V. CONCLUSION

In this paper¹ we proposed C^3P_o a new link-signature-based adversary detection mechanism for industrial wireless networks. Although link signatures are neither secret nor unforgeable, we argued that combining several individual measurements is a potential remedy. The evaluation shows that already few nodes significantly increase the detection performance and robustness.

REFERENCES

- [1] M. Demirbas and Y. Song, "An rssi-based scheme for sybil attack detection in wireless sensor networks," in *WOWMOM*, IEEE, 2006.
- [2] Y. Sheng, K. Tan, G. Chen, D. Kotz, and A. Campbell, "Detecting 802.11 mac layer spoofing using received signal strength," in *INFOCOM*, IEEE, 2008.
- [3] D. B. Faria and D. R. Cheriton, "Detecting identity-based attacks in wireless networks using signalprints," in *WiSe*, ACM, 2006.
- [4] H. Liu, Y. Wang, J. Liu, et al., "Authenticating users through fine-grained channel information," *IEEE Trans. Mobile Comput.*, 2017.
- [5] Z. Jiang, J. Zhao, X.-Y. Li, J. Han, and W. Xi, "Rejecting the attack: Source authentication for wi-fi management frames using csi information," in *INFOCOM*, IEEE, 2013.
- [6] N. Patwari and S. K. Kasera, "Robust location distinction using temporal link signatures," in *MobiCom*, ACM, 2007.
- [7] X. Zhang and E. W. Knightly, "CSIsnoop: Attacker inference of channel state information in multi-user WLANs," in *MobiHoc*, ACM, 2017.
- [8] X. He, H. Dai, W. Shen, and P. Ning, "Is link signature dependable for wireless security?" In *INFOCOM*, IEEE, 2013.
- [9] S. Fang, Y. Liu, and P. Ning, "Mimicry attacks against wireless link signature and new defense using time-synched link signature," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 7, 2016.
- [10] L. Sang and A. Arora, "Spatial signatures for lightweight security in wireless sensor networks," in *INFOCOM*, IEEE, 2008.
- [11] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *ACM SIGCOMM CCR*, no. 1, 2011.
- [12] A. Wulf, L. Underberg, and R. Kays, "Coverage range analysis of wireless technologies for industrial automation - system overview and performance evaluation," in *ICETE*, INSTICC, 2016.
- [13] S. Dietrich, G. May, O. Wetter, H. Heeren, and G. Fohler, "Performance indicators and use case analysis for wireless networks in factory automation," in *EFTA*, 2017.
- [14] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, 1982.
- [15] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, 1983.
- [16] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *OSDI*, vol. 99, 1999.

¹The work was partly funded by the German Federal Ministry of Education and Research under BMBF grant agreement no. 16KIS0222.