Amigo: Proximity-Based Authentication of Mobile Devices

Alex Varshavsky[1], Adin Scannell[1], Anthony LaMarca[2], and Eyal de Lara[1]

[1] Department of Computer Science, University of Toronto
{walex,amscanne,delara}@cs.toronto.edu
[2] Intel Research Seattle
anthony.lamarca@intel.com

Abstract. Securing interactions between devices that do not know each other a priori is an important and challenging task. We present Amigo, a technique to authenticate co-located devices using knowledge of their shared radio environment as proof of physical proximity. We present evaluation results that show that our technique is robust against a range of passive and active attacks. The key advantages of our technique are that it does not require any additional hardware to be present on the devices beyond the radios that are already used for communication, it does not require user involvement to verify the validity of the authentication process, and it is not vulnerable to eavesdropping.

1 Introduction

We envision that with the increased adoption of mobile devices, spontaneous communication between wireless devices that come within proximity of each other but lack a pre-existing trust relationship will become common. For example, patrons at a bar, guests at a party or conference participants may use their mobile phones to exchange private contact information over Bluetooth or WiFi. Consumers may use their mobile devices as electronic wallets to pay for tickets at the train station or groceries at the store. A user may take advantage of resources available in the environment by pairing their mobile phone to a public full-sized display and keyboard [3], or share music by pairing their phone to a friend's home entertainment system.

An important precondition for the widespread proliferation of spontaneous communication among wireless devices is securing these interactions against eavesdropping, impostors, and man-in-the-middle attacks, in which an attacker reads and inserts messages between two parties without either party knowing that the channel between them has been compromised. Obviously, users would not want their private contact or banking information to be overheard or tampered with by a malicious third party.

We refer to the problem of securing the communication between devices in proximity as *secure pairing*. Unfortunately, traditional cryptographic techniques, such as the Diffie-Hellman protocol [6], by themselves are not sufficient for securely pairing devices that spontaneously come into wireless contact. Whereas

J. Krumm et al. (Eds.): UbiComp 2007, LNCS 4717, pp. 253–270, 2007.

they provide a secure binding of keys to electronic identifiers, such as network addresses or device names, these techniques cannot guarantee that the two devices that the user holds in their hands are in fact the ones that are paired – an attacker hundreds of meters away with a directional antenna could be impersonating the device name or network address. What is required is a natural way to authenticate that the keys obtained through the cryptographic exchange belong indeed to devices that are within physical proximity.

This paper shows that it is possible to securely pair devices that come within close proximity by deriving a shared secret from dynamic characteristics of their common radio environment. This approach takes advantage of three observations. First, many mobile devices come equipped with radios that can sense their immediate radio environment. Second, devices in close proximity that simultaneously monitor a common set of ambient radio sources, e.g., WiFi access points or cell phone base stations, perceive a similar radio environment. For high frequency radio technologies, receivers only a few centimeters away may perceive different radio environments due to multi-path effects; however, these differences are generally small compared to differences perceived by receivers at larger distances. Third, due to environmental factors the radio channel varies in unpredictable ways over short time scales. For example, at a single location, signal strength from a cell phone base station fluctuates from one moment to the next, but devices in close proximity perceive similar fluctuations.

Together, these observations imply that it is possible for devices in close proximity to derive a common radio profile that is specific to a particular location and time. This paper shows that this profile can be used to securely pair devices in close proximity by using knowledge of their common radio environment as proof of physical proximity.

We describe Amigo, a technique that extends the Diffie-Hellman key exchange with verification of device co-location. Initially, the two devices perform a Diffie-Hellman key exchange in order to derive a shared-secret. After this exchange alone, it is not possible for either device to be sure whether it shares a secret with the other co-located device or with some potentially malicious third party. Next, both devices monitor their radio environment for a short period of time and exchange a signature of that environment with the other device. Finally, each device independently verifies that the received signature and its own signature are similar enough to conclude that the two devices are in proximity. The verification takes into consideration both transmissions received by the devices and perceived signal strength fluctuations.

An evaluation conducted using WiFi-enabled laptops shows that without requiring user interaction, Amigo can recognize an attacker located as close as 3 meters away. However, if the user is willing to create some localized entropy in the radio environment by, for example, walking or waving their hand in front of the antennas of the two co-located devices, Amigo can detect an attacker located as close as 1 meter away, and can defeat a powerful attacker that has surveyed the environment and has control over ambient radio sources.

Amigo has three advantages over existing solutions: (a) it requires no additional hardware to be present on the devices besides the standard wireless radio already available on most devices; (b) in most cases, it requires no user involvement (beyond specifying that the devices are to be paired); and (c) because devices determine co-location by listening to their radio environment, as opposed to transmitting, Amigo is immune to eavesdropping attacks.

2 Problem Definition and Threat Model

We define the problem of secure pairing of devices in close proximity as follows. Two devices that are located nearby (within 1 meter) to each other but do not know each other a priori need to establish a channel between them that is both secure and authentic. A secure channel implies that no eavesdropper may intercept and decrypt messages between the endpoints, while authenticity requires that both endpoints are able to confirm the identify of each other. We assume that the devices can communicate over compatible wireless radios (e.g., WiFi) and that neither additional out-of-band communication channels are required (e.g., ultrasound) nor is additional hardware present on the devices (e.g., accelerometers).

We assume the presence of an attacker that will try to pair with one or both of the legitimate devices. We assume that the attacker is located beyond the distance that separates the two legitimate devices, and can sense the wireless environment, inject new traffic, and replay packets. Moreover, we assume that the attacker could have surveyed the location where the two legitimate devices are attempting to pair. The attacker can use this knowledge to convince the legitimate devices that they are co-located by predicting the perceived signal strength of ambient radio sources at the location and time of pairing. In the most extreme case, we assume that the attacker both knows what packets were heard by the legitimate device and has access to distributions of signal strengths for each radio source as received by the legitimate device at the time and location of pairing. This is a best-case scenario for the attacker who, even with full control over ambient radio sources, would at best be able to transmit packets at known power levels and predict which packets were received by the legitimate device and at what signal strengths.

We consider two kinds of possible attacks: An *impostor attack* where the attacker succeeds in disabling one of the co-located devices and attempts to impersonate it; and a *man-in-the-middle attack* where the attacker attempts to pair with the two co-located devices simultaneously, and hides its presence by relaying authentication traffic between them.

3 Secure Pairing of Devices in Physical Proximity

In this section, we describe our algorithm to authenticate co-located devices using measurements of their shared radio environment as proof of physical proximity. Our solution is based on the observation that due to environmental effects it is

very hard to predict fluctuations in the radio environment at a specific location and at a specific time without being physically present at that location at that time. On the other hand, devices that are positioned in proximity not only tend to successfully decode radio transmissions from the same sources, but also perceive similar fluctuations in signal strength. We will show how this common radio environment can be used as a basis of an authentication scheme for co-located devices.

The Diffie-Hellman protocol allows two parties to create a shared secret key that can be used to secure future communications. While the protocol cannot be compromised by eavesdropping, it is susceptible to man-in-the-middle attacks by a third party. The protocol also does not provide any assurances as to the identity or the proximity of the devices that end up pairing. To both protect the protocol against man-in-the-middle attacks and to ensure that the pairing actually happens with a device in close proximity (as opposed to a far away attacker with a sensitive antenna), we extended the Diffie-Hellman key exchange with a *co-location verification* stage.

In our scheme, after the two devices perform a Diffie-Hellman key exchange, each device monitors the radio environment for a short period of time and generates a signature, which includes a sequence of packet identifiers[1] and the signal strength at which the packets were received. This signature is then transmitted to the other device over the secure channel via a commitment scheme intended to secure the pairing against a man-in-the-middle attack. Finally, each node independently verifies that the received signature and its own signature are similar enough to conclude that the two devices are co-located. At the end of the verification, each device either accepts the signature or rejects it. Because the signatures are used only to validate the keys being exchanged, but not as the basis for encryption, the signatures do not have to remain secret once the authentication takes place. The only requirement on the signatures is that they have to be hard to guess during the authentication phase.

Next, we present our co-location verification algorithm and describe our commitment scheme which is designed to prevent a man-in-the-middle attack.

3.1 Co-location Verification Algorithm

The problem of co-location verification can be described as follows. Given a signature captured locally by a device A, and a signature received from a device B, A needs to reach a binary decision as to whether B is co-located or not. For A to conclude that B is co-located, the signature received from B should be sufficiently similar to the signature captured locally. The verification algorithm has four stages: temporal alignment, slicing, feature extraction and classification. This process is shown in Figure 1.

Since the two devices capture packets locally, they may have started capturing packets at slightly different moments. To meaningfully compare sequences of packet identifiers and signal strength measurements, they need to be temporally

[1] We use hashes of packet headers as packet identifiers.

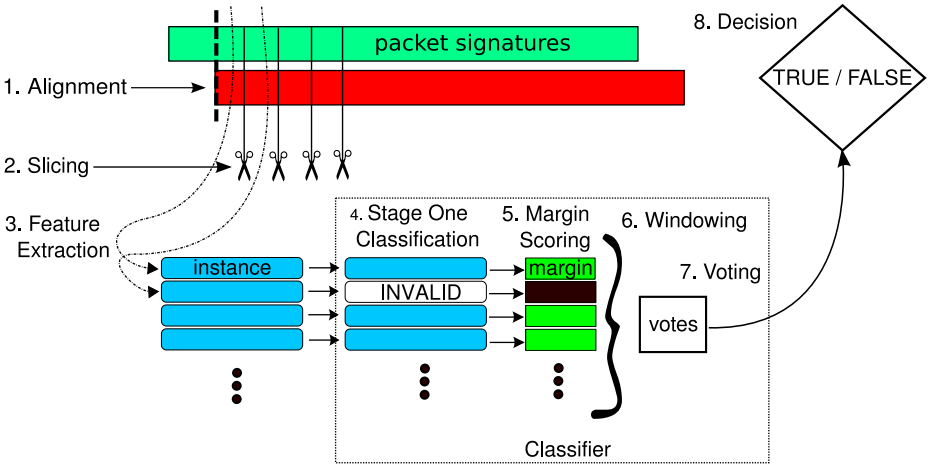


Fig. 1. The complete co-location verification process

aligned. The verification algorithm begins by aligning the packet sequences using the first common packet identifier and discarding all preceding elements in each sequence.

Once the signatures are temporally aligned, we slice them into smaller consecutive subsequences of a fixed timespan, known as *segments*. Slicing allows the verification algorithm to first give a similarity score to each pair of aligned segments and then combine these scores into a final classification decision. Using segments of one second in length typically allows enough packets to be observed for good performance while still keeping the protocol responsive.

The verification algorithm then extracts a set of features from each of the resulting aligned segment pairs. Each feature captures a particular relationship between the two segments to be used by the classification algorithm. For example, percentage of packets that are common to both segments is a useful feature because a higher percentage is associated with a higher likelihood that the two devices are co-located. We describe the set of features used by our classifier in Section 4.2. We further refer to a set of features extracted from a pair of segments as an *instance*.

Finally, we feed the set of instances (one for each segment) into a classifier, which gives a decision as to whether the two signatures have been captured by co-located devices.

The Classifier. We distinguish instances generated by co-located devices from instances generated by non co-located devices using a two stage boosted binary stump classifier. The first stage was added in order to filter noisy data before the more complex binary stump classifier, allowing effective training with less data. During the first stage, instances that have less than a minimum percentage of packets in common are marked as invalid; these are treated in a special way at the end of the second stage. Instances with a low percentage of packets in common (which occur much more commonly with non co-located pairs) were

$$
\begin{aligned}
Margin(i) &= 2.64 \times x_1(i) + 1.79 \times x_2(i) + \ldots + 1.16 \times x_9(i) + 2.61 \times x_{10}(i) \\
x_1(i) &= \begin{cases} 1 & \text{if } i.feature_A \leq 4.31625 \\ -1 & \text{otherwise} \end{cases} \\
&\ldots \\
x_{10}(i) &= \begin{cases} 1 & \text{if } i.feature_B \leq 5.297 \\ -1 & \text{otherwise} \end{cases}
\end{aligned}
$$

Fig. 2. A sample margin calculation for an instance i during the second stage of classification. The same feature may appear within multiple x_i definitions.

found to have very high classification error. We found experimentally that a threshold of 75% works well.

In the second stage, valid instances are assigned a score, referred to as a *margin*. A margin is derived by evaluating a set of simple functions on an instance and combining the results in a weighted-sum. A sample margin calculation is shown in Figure 2. A larger positive margin indicates more confidence that the devices are co-located. A lower negative margin indicates more confidence that the devices are not co-located. A margin near zero indicates a lack of confidence about the decision.

Finally, the classification algorithm aggregates a window of margins and makes a prediction. Windowing allows the classifier to tolerate a lower degree of accuracy with each individual instance classification. The decision for each window is made with a simple voting scheme. We convert margins into votes based on an adjustable margin threshold; margins that are higher than the threshold are converted to TRUE votes and margins that are below the threshold are converted to FALSE votes. Instances that are marked as invalid do not contribute any vote. At the end, if the window contains a majority of TRUE votes, the devices are classified as co-located. Excessive invalid instances or a majority of FALSE votes will cause the classified to be classified as not co-located. In Section 4.4, we show that the margin threshold of 4 works well in practice and, in Section 4.3, we evaluate the effect of the window size on the classification accuracy.

In order to train the classifier, appropriate features, constants and weights for the margin calculation must be selected. This process is discussed in Section 4.2.

3.2 Dealing with a Man-in-the-Middle Attack

In order to deal with a man-in-the-middle attack, our algorithm employs a simple commitment scheme. The trace collection is broken into short periods of a fixed duration during which a device captures one block of data. The devices are required to exchange the blocks at the end of each time period, otherwise the pairing is rejected. Before sending a block, a device concatenates the block with a hash of its Diffie-Hellman session key and its device identifier, encrypts the result using a nonce value and sends this encrypted block to the other device. Concatenation of the session key is required to prevent the attacker from simply forwarding blocks back and forth between the co-located devices as explained

below, and concatenation of the device identifier prevents the attacker from simply "mirroring" the messages. After all blocks have been transferred, both devices exchange the set of nonces required to decrypt the sequence of encrypted blocks and verify the hashes of the session keys and device identifiers.

At the end of each time period, the attacker is required to supply a block. Since the attacker cannot decrypt received blocks until the end of the collection process, he has only two choices. The attacker can either pass on the encrypted block received from a co-located device or can generate a new block with its own session key. Since the session keys between the attacker and each co-located device are different, simply passing the encrypted blocks between devices will not allow the attacker to pair with either of the devices. In Section 4, we show that trying to forge new packets based on the radio environment is also likely to be fruitless, unless the attacker is very close to the co-located devices.

This scheme is equivalent to a fixed-delay interlock protocol [15]. The nonce used to decrypt the blocks can be sent one collection period after the encrypted blocks. However, unlike the fixed-delay interlock protocol, it is not necessary to use the delays to simply detect a man-in-the-middle; after the relevant time period has passed, each block becomes useless, as they are strictly time-dependent. The attacker can not extract useful information from any block in order to pass it to a target when it is required. The nature of the secrets implicitly detects a man-in-the-middle by forcing him to generate fake traces.

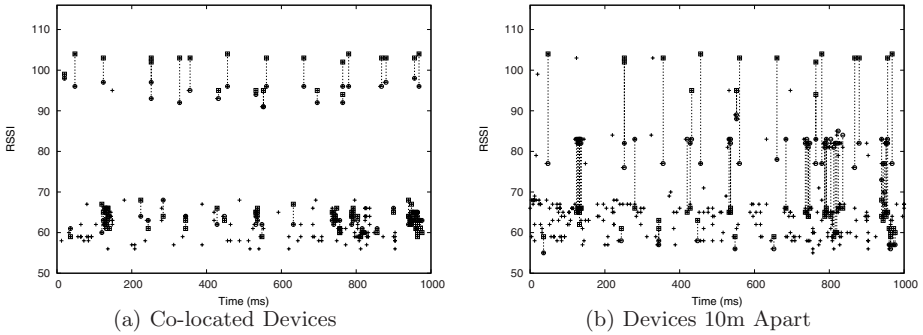
4 Evaluation

In this section, we first discuss our data collection and training procedures, then we proceed to evaluate the performance of Amigo under various conditions. First, we test the basic configuration, similar in nature to our training configuration, but using data collected at a different time and place. We then test the effect of obstacles between the attackers and the co-located devices. Subsequently, we experiment with more sophisticated attacker scenarios, and explore having the user generate localized entropy in order to improve accuracy. Finally, we briefly discuss current limitations of this evaluation.

4.1 Data Collection

We collected WiFi traces using a testbed consisting of 6 laptop computers (3 ThinkPad, 2 Dell and 1 Toshiba), all equipped with Orinoco Gold WiFi cards. WiFi is a practical technology for evaluating Amigo, since it is increasingly prevalent. At 2.4GHz, a few centimeters can make a difference in a multi-path channel, but in our experience the differences in the radio environment observed by nodes separated by greater distances tend to dominate these smaller dissimilarities.

Two laptops that were playing the role of the co-located devices were positioned 5 centimeters apart in an opposite orientation, so that their WiFi cards would be located as close as possible. Four additional laptops positioned 1 meter, 3 meters, 5 meters and 10 meters away from the co-located laptops



(a) Co-located Devices (b) Devices 10m Apart

Fig. 3. The received signal strength for packets heard by two devices in a 1 second period. Packets marked as "+" were received by only one laptop. Lines connect packets heard by both devices.

were playing the roles of malicious devices. We felt that these distances would provide a good indication of the ability of the algorithm to differentiate adversaries at various distances. Whereas most of our experiments only consider the devices positioned 5 centimeters apart to be co-located, we explore stretching this distance to 1 meter in Section 4.6.

We wrote a script that simultaneously switches the WiFi cards on all laptops into a monitor mode and captures packets overheard by the cards. The WiFi drivers on all laptops were modified not to discard corrupted packets, but simply mark the packets as corrupted.

An active scan for WiFi access points in the lab environment where data collection took place reveals 11 access points on average. With a 10 minute trace taken in the afternoon, each laptop captured between 30 thousand and 50 thousand packets (including all WiFi beacons, etc.) and heard between 45 and 58 unique transmitters. The majority of transmitters heard in this environment were not loquacious - each laptop captured less than 100 packets from most sources.

Figure 3 shows the packets received by two co-located devices and non co-located devices over a period of 1 second . The y axis represents the Received Signal Strength Indicator (RSSI) value associated with each packet. The packets marked by "+" were received by only one of the two laptops, while packets connected by lines were received by both laptops. For the co-located pair, during this time period approximately 85% of the packets received were common to both. For the distant pair, only 40% of the packets were common. Comparing the figures also immediately makes apparent the similarity of RSSI values for the co-located pair.

4.2 Training the Classifier

As discussed in Section 3.1, the classifier must be trained. For this purpose, we used a MultiBoost [24] algorithm with decision stumps (single node decision trees) as its base learner. The MultiBoost algorithm is a decision committee

Table 1. A relevant subset of features extracted from aligned segments. The sequences of RSSI values for the N common packets in each instance are represented by a_i and b_i, $0 \leq i \leq N$.

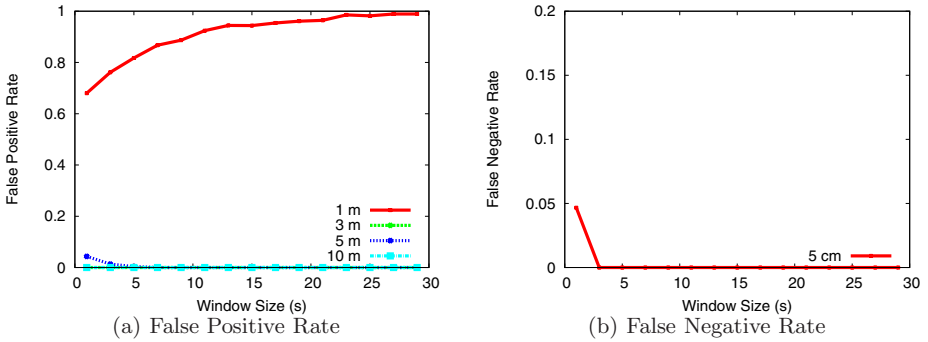
Feature Name	Feature Definition	Description
signal:abs	$\frac{\sum_1^N \lvert a_i - b_i \rvert}{N}$	The mean absolute difference between received signal strength measurements.
signal:eucl	$\sqrt{\sum_1^N (a_i - b_i)^2}$	The euclidean difference between received signal strength vectors.
signal:exp	$\frac{\sum_1^N e^{\lvert a_i - b_i \rvert}}{N}$	The mean exponential of the difference between signal strength measurements.
signalexp:diff:eucl	$\sqrt{\sum_2^N \left(e^{(a_i - a_{i-1})} - e^{(b_i - b_{i-1})}\right)^2}$	The euclidean difference between exponential signal strength deltas.

technique that combines AdaBoost [7] with wagging. This approach has been shown to be more effective in reducing error than either of its constituent techniques [24]. The MultiBoost algorithm selects the appropriate set of weighted linear classifiers that are used for the margin calculation for each valid instance.

For training, we captured 10 minutes worth of packets on all 6 laptops. We aligned and sliced all packet sequences captured by each pair of laptops in our testbed, and then extracted features from all pairs of sequences that included the first co-located laptop. We trained our classifier using 596 instances from co-located devices and 2279 instances from non co-located devices.

To evaluate the effectiveness of the classifier, we investigate its performance in terms of false positive and false negative rates. False positives occur when the algorithm predicts that the devices are co-located when they are in fact not, and the *false positive rate* is the number of false positives divided by the total number of non co-located instances. False negatives occur when the algorithm predicts that the devices are not co-located when they in fact are, and *false negative rate* is the number of false negatives divided by the total number of co-located instances. Note that in general, reducing false positives is ultimately more important that reducing false negatives because confusing a malicious device for a co-located device is a more serious flaw than not admitting the co-located device and requiring the user to wait for a longer period before the devices get paired together.

Training the classifier with all available features achieves 23 false negatives out of 596 true instances and 50 false positives out of 2279 instances. Although we constructed dozens of features, only four were selected by the MultiBoost algorithm during training: the mean absolute difference in signal strength (*signal:abs*), the mean exponential difference in signal strength (*signal:exp*), the euclidean difference between signal strength vectors (*signal:eucl*) and the euclidean difference between exponential signal strength deltas (*signalexp:diff:eucl*). These are the features shown in Table 1.



(a) False Positive Rate (b) False Negative Rate

Fig. 4. Co-located devices are 5cm apart. The attackers are 1m, 3m, 5m and 10m away.

4.3 The Base Case

After training the classifier, we collected a second set of data for testing our technique. To allow time for the radio environment to evolve, the testing data set was collected two months after the training data set was collected. To prevent the classifier from recognizing any anomalies with the particular physical arrangement of devices, the collection setup was moved from one end of our lab to the other (approximately 10 meters). Besides changing the location and time of collecting the testing data, the experimental setup was left unchanged – two co-located laptops were positioned 5 centimeters apart and 4 other laptops were positioned 1 meter, 3 meters, 5 meters and 10 meters away from the co-located laptops. All laptops were positioned at the same height and the attackers had a line of sight to the co-located devices, a likely best-case scenario for the attackers.

We tested the ability of our classifier to authenticate co-located devices and reject non co-located device as a function of the classifier's window size. Since we are using 1 second segments, the window size represents the amount of time, in seconds, that a user needs to wait to authenticate the devices. Figure 4(a) plots the false positive rates of attackers trying to authenticate while located 1 meter, 3 meters, 5 meters and 10 meters away, while Figure 4(b) plots the false negative rate of not authenticating a co-located device located 5 centimeters away. The results show that in 5 seconds both the false negative rate falls to 0, meaning that all attempts of the co-located device to authenticate are successful. In the same 5 seconds, the false positive rate falls to 0 for the attackers located 3 meters, 5 meters and 10 meters away, meaning that the attacks initiated from at least 3 meters away were all unable to fool the system. Unfortunately, the attacker device that is located 1 meter away is able to convince the other device that it is co-located. Since more than 60% of instances from the attacker 1 meter away are incorrectly classified as co-located, increasing the window size results in aggregation of a larger proportion of these instances in every window and consequentially a more consistent misclassification of the attacker.

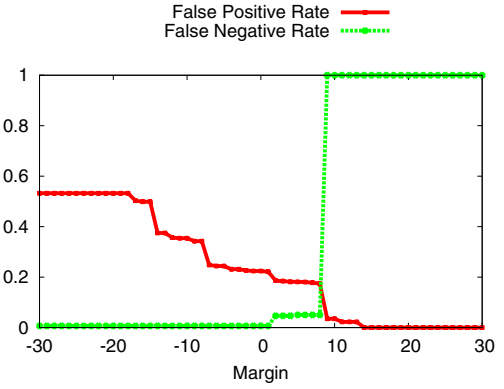


Fig. 5. The false positive and false negative rates for different margin thresholds

To prevent attackers as close as 1 meter from succeeding, we explored a technique as we proposed [23] to generate localized entropy in the radio environment. The user that suspects that a possible attacker may be nearby simply needs to wave their hand in front of the antennas of the two mobile devices during the pairing. This motion will generate unique fluctuations in the radio environment perceived only by the co-located devices. We tested the effect of hand waving on the ability of the 1 meter away attacker to authenticate with a device. In this case, the false positive rate of the attacker falls to 0 within 5 seconds. Hand waving is a natural and non-burdensome action for users to perform in order to pair devices. To provide a more secure pairing, users can be encouraged to move around and use their preferred hand motions as if invoking some form of personal sorcery by casting a pairing spell.

4.4 The Effect of Margin Threshold

Recall that the classifier assigns a margin to each valid instance which is compared to a threshold in order to determine a vote. Increasing this threshold makes the classifier less likely to accept the devices as co-located, increasing the chance of a false negative but also decreasing the chance of a false positive. Reducing the threshold has the opposite effect. Figure 5 plots the false negative and false positive rates for all individual segments as a function of this margin threshold. As the margin threshold grows, fewer segments belonging to impostors are authenticated and as a consequence the rate of false positives falls. The jumps in the false positive and false negative rates result from the finite number of values that the margin can take, since it is the sum of a finite number of constants. Setting the margin threshold to a value beyond 8 results in no attackers being authenticated, but also in no co-located devices being authenticated, in which case the false negative rate rises to 1. The plateau between the margin threshold of 2 to 8 seems to consistently strike a good balance between false positives and false negatives. We used margin threshold of 4 for all our experiments.

Table 2. The effect of different materials on the ability of the attacker to authenticate with a device

Obstruction	False Positive Rate
None (1m)	0.81
Drywall (10cm)	1.00
Human (1m)	0.12
Concrete Wall (30cm)	0.00

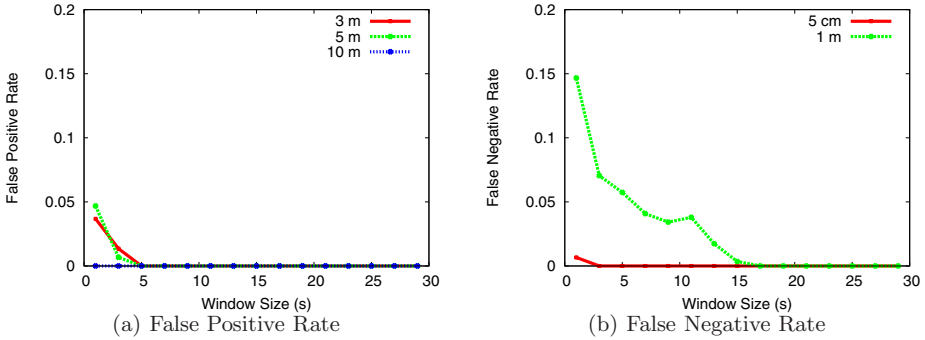
4.5 The Effect of Obstacles

In all the experiments described above, the attacker had a clear line of sight to the co-located devices. In reality, that might not be the case as different kinds of materials may block or obstruct the path between the attacker and the target. We looked at the effect of three common materials blocking the line of sight between the attacker and co-located devices, including drywall, concrete and a human body. Table 2 summarizes our findings in terms of false positives.

For the attacker 1 meter away and with a 5 second window, the false positive rate is about 80% as was shown in Figure 4(a). In contract, when the attacker is separated from the co-located devices by two sheets of gypsum drywall, a wall about 10 centimeters thick, the false positive rate climbs to 1. This has the implication that drywall does not protect users from an attacker who is immediately behind the wall. This is in line with other signal propagation studies that have shown that dry wall does not have a profound effect on radio signal propagation in the 2.4GHz range [20]. When a human being is standing between the the attacker and the two co-located laptops located 1 meter away, the false positive rate falls to just above 10%. This is encouraging, as it means that humans, being basically bags of water, by just the sheer blocking of the authentication with their buddies, may make it significantly more secure. Finally, when a 30 centimeter-thick concrete wall separates the attacker from the co-located devices, the false positive rate is 0. When passing through a concrete wall, radio signals attenuate strongly enough to make it extremely hard for the simple attacker to authenticate without more sophisticated attacks.

4.6 Stretching Co-location

Up until now, we used a classifier trained with data that indicated that two devices were co-located if they were 5 centimeters apart. In this section, we study the effects of extending the notion of co-location to include devices located up to 1 meter away. We retrained our classifier on the same training data, but this time we marked the segments belonging to the device located 1 meter away as also co-located. Figures 6(a) and 6(b) plot the false positive and false negative rates as a function of time the user needs to wait to authenticate the devices. The results show that after 5 seconds the false positive rate falls to 0 for all attackers located 3 meters or more away, while the false negative rate falls to 0.05. This means that in 5% of cases a user will not succeed to pair co-located devices in



(a) False Positive Rate (b) False Negative Rate

Fig. 6. Co-located devices are up to 1m apart. The attackers are 3m, 5m and 10m away.

5 seconds from the first attempt, and will need to retry again. However, waiting for 20 seconds always results in a correct pairing.

4.7 Sophisticated Attacks

A powerful attacker could have surveyed the location where the two legitimate devices are attempting to pair, and could attempt to use this knowledge to convince the legitimate devices that he is currently present at that particular location. We implemented a simulated powerful attacker for the purpose of evaluating the robustness of our system under such a threat.

We conducted an experiment where the attacker had access to a distribution of received signal strength measurements for each radio source, sampled by the target device itself at the pairing location only a few hours prior to the current authentication. During the authentication, when the attacker receives a packet from a radio source that he has observed before, he substitutes the signal strength value in this packet with a sample from the recorded distribution of packets previously observed at the pairing location from that transmitter. Whenever the attacker receives a packet from a source that he has no distribution for, the attacker has a choice of either pretending he never received the packet in the first place, thereby potentially decreasing the percent of common packets if the target device did get this packet, or simply leaving the signal strength in the packet as is. Experiments showed that not discarding these packets is beneficial to the attacker.

Figure 7 shows that attackers located 1 meter and 5 meters away can successfully authenticate in 15% and 45% of the cases using 5 second windows, respectively. The adversary positioned 5 meters away actually performed better in our experiments than the laptop positioned 3 meters away, due to the fact that the laptop at a distance of 5 meters generally shared a slightly larger number of packets with the target. The laptops positioned 3 meters and 10 meters away were not able to authenticate because a large portion of their instances

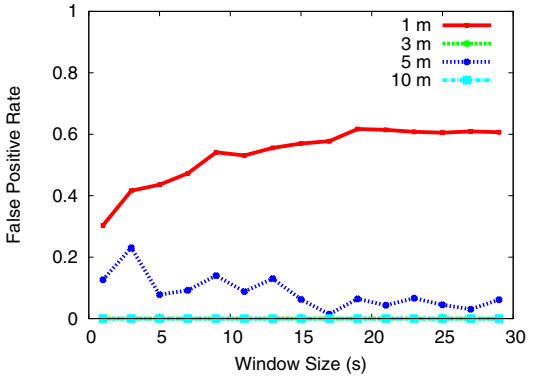


Fig. 7. False positive rate for a simulated attacker who samples from the distribution of co-located devices and then uses that signal strength to impersonate one of the devices

were rejected due to insufficient common packets before moving to the second stage of classification.

In order to defend against the attacker who has gone to the measure of rigorously surveying the environment in this way, we propose to use the hand waving enhancement discussed earlier. Even equipped with a location-specific distribution, the system has a false positive rate of 0 within all 5 second windows of the experiment for all attackers.

In a worst-case scenario, an attacker could also be powerful enough to have complete control over the radio environment. We assume that such an attacker has injected every packet into the environment, and that he knows exactly what packets the target device receives. Also, for our simulation, the attacker is equipped with an oracle, that allows the attacker to sample from the distribution of signal strength values as perceived by the target device over the duration of the experiment.

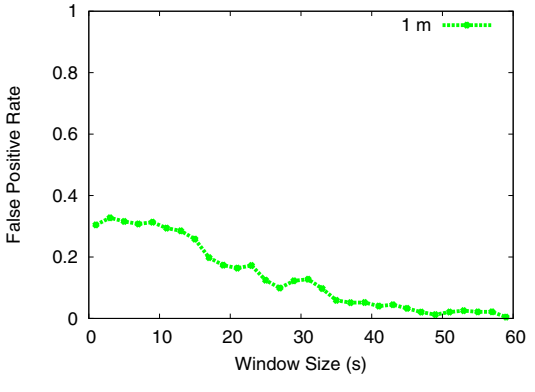


Fig. 8. Hand waving allows Amigo to prevent even the most sophisticated attack, when the attacker can predict the signal strength of the `current` distribution of packets

We tested whether hand waving will prevent this attacker from authenticating with the target device in this case. Figure 8 shows the false positive rate of authentications with the oracle attacker. The figure shows that with hand waving, after 5 seconds, the attacker is able to authenticate successfully in only 30% of the cases. If the user is willing to pair the devices for 60 seconds, the false positive rate falls to 0%! This is encouraging given the small likelihood of encountering an attacker this powerful.

4.8 Discussion

So far, we evaluated Amigo using homogeneous WiFi cards. Although using WiFi cards with different chipset designs or antenna configurations will have an effect on received signal strengths, we believe that it will be possible to generalize Amigo by normalizing the received signal strength values and make Amigo antenna and chipset agnostic. As a preliminary test of this hypothesis, we hooked an external omnidirectional antenna onto the WiFi card of one of the co-located laptops and collected a 10 minute trace. Using a simple gain-correction, wherein RSSI measurements of the antenna-equipped laptop are adjusted by a small, constant amount, Amigo was still able to achieve a low level of false negatives with zero false positives after only a few seconds using the standard classifier presented in Section 4.3.

Hardware heterogeneity is not the only factor that requires further exploration. The success of our technique may also depend on physical or network conditions. For example, the location of our experiments had moderate to heavy WiFi usage. It would be interesting to evaluate the system in quieter environments, such as at a small café. All of our experiments were also conducted on flat, stable surfaces. Experimenting with small devices held by shaky hands is a topic for our future research.

5 Related Work

SWAP-CA [21] is a specification that gives users a way to associate devices by pressing a button on two devices simultaneously, but does not provide security. In Bluetooth, users pair devices by providing each device with a secret PIN number. While the PIN provides for device authentication, it requires active user involvement and interaction with both devices. Moreover, Bluetooth pairing has been shown to be susceptible to attack by eavesdroppers equipped with sensitive directional antennas, which enable attackers to breach the security of the system from more than a mile away [5,17]. LoKey [14] uses SMS messages as an out-of-band channel to authenticate a key exchanged over the Internet. While this approach is secure, SMS delivery is slow and may incur monetary cost.

Physically shaking two devices together for authentication has recently received significant attention in the research community. Smart-It [9] used common readings from accelerometers to establish an association between devices shaken at the same time. Mayrhofer and Gellersen extended this technique to

provide secure authentication between the shaken devices [12]. Both of these techniques use the accelerometer readings as the basis of the authentication. In Shake Them Up! [4], two devices establish a shared secret over an anonymous broadcast channel by taking turns transmitting parts of the key. Shaking the devices randomizes the reception power of their packets by a potential eavesdropper and makes it hard for attackers to exploit power analysis to break the channel anonymity. Unfortunately, this approach is vulnerable to attack by an eavesdropper that exploits the differences in the baseband frequencies of the two radio sources, which result from differences in their crystal clock oscillators, to differentiate between packets sent by the two transmitters. In general, shaking techniques are easy for users to understand and when accelerometers are available, provide intuitive and reliable device pairing. The obvious drawback with shaking techniques is that there are objects such as ATM machines and vending machines that are too large or too heavy to be shaken vigorously. This inspired our hand waving technique as it does not require both objects to be shaken together and only requires hands to be waved or shaken near the antennas of the two co-located devices to generate localized entropy.

Numerous research projects have suggested the use of physically constrained channels as a means of establishing secure association between devices in close proximity. Some examples include the use of a direct electric contact [19], infrared beacons [2,18], ultrasound [11], and laser beams [10]. Unfortunately, physically constrained channels often require extra hardware (e.g., an extra cable), and can be susceptible to attacks by sensitive receivers that can detect dim signal refractions and reflections [5,17].

Another technology that can be used for secure device pairing is Near Field Communication or NFC. Unlike radio-transmission based wireless communication like 802.11 and GSM, NFC transmits data via inductive loading. This limits the working range of NFC links to a few centimeters [1]. While obviously a good fit for many use cases, NFC is not without issues: like traditional radio transmissions, the range of the inductive loading can be drastically increased by eavesdropping with a large antenna – a large loop of wire in the case of NFC. It is also the case that for cultural or hygiene reasons, there are situations in which the "almost touching" nature of NFC may be inappropriate and for which the notion of proximity may be better suited by a larger distance. Lastly, NFC does add additional cost, size and weight to a mobile device in addition to the far-field communication already present.

Another solution to establishing trust between mobile devices is to use a public key infrastructure. In this case, every mobile device is uniquely named and certified by a trusted authority. Even if the effort is spent to grant every device a unique and certified name, pairing may still require significant user involvement since there may be multiple nearby devices to choose from.

Several projects proposed to delegate the verification of whether the two intended devices have been paired to the user. McCune *et al.* [13] and Saxena *et al.* [16] proposed to use the visual channel for verification, while Goodrich *et*

al. [8] proposed to use the audio channel. Uzun *et al.* [22] recently performed a comparative evaluation of a number of user-driven verification methods.

In contrast, we propose to use the common radio environment as a basis to the shared secret between co-located devices – if two devices perceive a similar radio environment they are probably very close to each other. A key advantage of our approach is that it makes use of the existing radio interfaces already present on mobile devices and does not require any additional hardware. The approach is also automatic and does not require user involvement to verify the correctness of the pairing.

6 Conclusions

In this paper we showed that it is possible to securely pair devices that come within close proximity by using knowledge of dynamic characteristics of their common radio environment as proof of physical proximity. We introduced Amigo, an algorithm that extends the Diffie-Hellman key exchange with verification of device co-location. Amigo has three key advantages: it does not requirement additional hardware beyond the wireless interface used for normal communication, it does not require user involvement to verify the pairing, and it is not susceptible to eavesdropping.

We evaluated Amigo using WiFi-enabled laptops and showed that within 5 seconds it is possible to recognize attackers located as close as 3 meters away from the co-located laptops. However, if the user is willing to wave their hands in front of the antennas of the two co-located devices in order to generate some localized entropy, Amigo is resilient to an attacker located 1 meter away, even in the unlikely case of an attacker that controls all ambient radio sources and is using the signal strength values from a distribution collected at the exact location of the current pairing.

References

1. Near field communication (nfc), `http://www.nfc-forum.org/resources/faqs`
2. Balfanz, D., Smetters, D., Stewart, P., Wong, H.: Talking to strangers: Authentication in ad-hoc wireless networks. In: Proc. Network and Distributed Systems Security Symposium (2002)
3. Barton, J.J., Zhai, S., Cousins, S.: Mobile phones will become the primary personal computing devices. In: IEEE Workshop on Mobile Computing Systems and Applications, April 2006, IEEE, Los Alamitos (2006)
4. Castelluccia, C., Mutaf, P.: Shake them up!: a movement-based pairing protocol for cpu-constrained devices. In: Proc. of MobiSys, pp. 51–64 (2005)
5. Cheung, H.: How to: Building a bluesniper rifle - part 1 (March 2005), `http://www.tomsnetworking.com/2005/03/08/how_to_bluesniper_pt1`
6. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Transactions on Information Theory, 644–654 (1976)
7. Freund, Y., Schapire, R.: Experiments with a new boosting algorithm. In: Proc. of International Conference on Machine Learning, pp. 148–156 (1996)

8. Goodrich, M., Sirivianos, M., Solis, J., Tsudik, G., Uzun, E.: Loud and clear: Human-verifiable authentication based on audio. In: Proc. of IEEE Internation Conference on Distributed Computing Systems, IEEE Computer Society Press, Los Alamitos (2006)
9. Holmquist, L.E., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M., Gellersen, H.W.: Smart-its friends: A technique for users to easily establish connections between smart artefacts. In: Abowd, G.D., Brumitt, B., Shafer, S. (eds.) Ubicomp 2001: Ubiquitous Computing. LNCS, vol. 2201, Springer, Heidelberg (2001)
10. Kindberg, T., Zhang, K.: Secure spontaneous device association. In: Dey, A.K., Schmidt, A., McCarthy, J.F. (eds.) UbiComp 2003. LNCS, vol. 2864, Springer, Heidelberg (2003)
11. Kindberg, T., Zhang, K.: Validating and securing spontaneous associations between wireless devices. In: Boyd, C., Mao, W. (eds.) ISC 2003. LNCS, vol. 2851, Springer, Heidelberg (2003)
12. Mayrhofer, R., Gellersen, H.: Shake well before use: Authentication based on accelerometer data. In: Fifth International Conference on in Pervasive Computing (2007)
13. McCune, J., Perrig, A., Reiter, M.: Seeing-is-believing: Using camera phones for human-verifiable authentication. In: Proc. of IEEE Symposium on Security and Privacy, pp. 110–124. IEEE Computer Society Press, Los Alamitos (2005)
14. Nicholson, A.J., Smith, I.E., Hughes, J., Noble, B.D.: Lokey: Leveraging the sms network in decentralized, end-to-end trust establishment. In: Fishkin, K.P., Schiele, B., Nixon, P., Quigley, A. (eds.) PERVASIVE 2006. LNCS, vol. 3968, Springer, Heidelberg (2006)
15. Rivest, R.L., Shamir, A.: How to expose an eavesdropper. Commun. ACM 27(4), 393–394 (1984)
16. Saxena, N., Ekberg, J., Kostiainen, K., Asokan, N.: Secure device pairing based on visual channel. In: Proc. of IEEE Symposium on Security and Privacy, IEEE Computer Society Press, Los Alamitos (2006)
17. Shaked, Y., Wool, A.: Cracking the bluetooth pin. In: Proc. of Mobisys (2005)
18. Smetters, D., Balfanz, D., Durfee, G., Smith, T., Lee, K.: Instant matchmaking: Simple, secure virtual extensions to ubiquitous computing environments. In: Dourish, P., Friday, A. (eds.) UbiComp 2006. LNCS, vol. 4206, Springer, Heidelberg (2006)
19. Stajano, F., Anderson, R.J.: The resurrecting duckling: Security issues for ad-hoc wireless networks. In: Malcolm, J.A., Christianson, B., Crispo, B., Roe, M. (eds.) Security Protocols. LNCS, vol. 1796, Springer, Heidelberg (2000)
20. Stone, W.C.: NIST Construction Automation Program Report No. 3: Electromagnetic Signal Attenuation in Construction Materials (NISTIR 6055), National Technical Information Service, Washington (October 1997)
21. SWAP-CA. Shared Wireless Access Protocol (Cordless Access) Specification (SWAP-CA), Revision 1.0, The HomeRF Technical Committee (17 December 1998)
22. Uzun, E., Karvonen, K., Asokan, N.: Usability study of secure pairing methods. Technical Report 2007-02, Nokia Research Center (January 2007)
23. Varshavsky, A., LaMarca, A., de Lara, E.: Enabling secure and spontaneous communication between mobile devices using common radio environment. In: IEEE Workshop on Mobile Computing Systems and Applications (HotMobile) (February 2007)
24. Webb, G.: Multiboosting: A technique for combining boosting and wagging. Machine Learning, 159–196 (2000)