Proximity-based Authentication of Mobile Devices

Adin Scannell, Alex Varshavsky, Anthony LaMarca, Eyal de Lara
Introduction

- Mobile devices are widely used
- We want using them to be easy, ubiquitous and secure
- Spontaneous communication
Secure Pairing

- Diffie-Hellmen is not enough
  - Provides key/device binding but which device?
- Use dynamic characteristics of shared physical environment
  - Devices already have radio equipment
  - Co-location -> similar environment
  - Hard to predict future conditions
Amigo

* Extends Diffie-Hellman with verification of device co-location
  * Use D-H to establish shared secret
  * Monitor radio traffic to build a signature
  * Compare partners signature with yours
Amigo, continued

- Benefits of Amigo
  - uses existing hardware
  - little to no user involvement
  - ease-dropping proof
Threat Model

- Two devices are near each other with no a priori knowledge of each other and want a secure, authentic communication channel

- Assume adversary:
  - further from legitimate devices than they are from each other
  - can monitor environment, transmit data and reply old packets
  - has previously surveyed environment
Attacks

- Impostor attack
- Man in the middle attack
Secure Pairing, implementation

- D-H prevents eavesdropping, not man in the middle
  - no guarantee of authenticity
- Co-location verification stage
  - prevents against man-in-the-middle if attacker is further away
Co-location Verification

- Capture packet identifiers and signal strength for each
- Transmitted via secure channel
  - commitment scheme
- Verify signature and local observations are “close enough”
- Each party and accept or reject
Determining Co-location

- Four stages:
  - temporal alignment
  - slicing
  - feature extraction
  - classification
Feature extraction

• Two-stage boosted binary stump classifier
  • First stage: rejects instances below a given threshold
  • Second stage: assign a scored called a margin
• Large positive results indicate co-location, lower negative scores indicate non-co-location, near zero indicates lack of confidence either way
Classification

- Each instance turns into a vote
- Margin above adjustable threshold $\rightarrow$ True vote
- Margin below adjustable threshold $\rightarrow$ False vote
- Invalid instances do not count as vote
- If the majority of votes are True $\rightarrow$ devices are co-located
- Majority of votes are false or excessive invalid instances $\rightarrow$ not co-located
Preventing MitM Attacks

- Data is exchanged on known intervals
  - If one device doesn’t send at the end of the interval, pairing fails
  - Data includes: signature of captured packets, hash of session key, device ID
- Data is encrypted with nonce value
- After all data for pairing is sent, nonces are exchanged and each starts verification process
Since data must be sent at the end of each time period, attacker has two choices:

- Forward the same data as the other device in the pairing
- Generate new data with its own session key

Similar to fixed-delay interlock protocol [Rivest and Shamir, 1984]
Test Results

- Using WiFi at 2.4 GHz
- Pairing devices 5 cm apart
- Attackers at 1m, 3m, 5m and 10m
- Lab had 11 access points on average
  - In 10-minute span detected: 30k - 50k packets, 45 - 58 unique transmitters, most transmitters sent < 100 packets
Training Classifier

- MultiBoost algorithm with decision stumps
  - decision committee technique combining AdaBoost with waging.
  - sets appropriate set of weighted linear classifiers for margin calculation of valid instances
- Used 596 instances from co-located devices and 2279 from non co-located devices (10 minutes from each of the 6 devices)
Training Results

- False-positive: devices deemed co-located that are not
- False-negative: devices deemed non co-located that are co-located
- False-positive is much more harmful than false-negative
- Training yielded 23 false-negatives of the 596 co-located instances and 50 false positives out of 2279 non co-located instances
Training Results, continued

- 4 features selected by MultiBoost:
  - Absolute difference in signal strength
  - Mean exponential difference in signal strength
  - Euclidian difference between signal strength vectors
  - Euclidian difference between exponential signal strength deltas
## Training Results, continued

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Feature Definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>signal:abs</td>
<td>$\frac{\sum_{1}^{N}</td>
<td>a_i - b_i</td>
</tr>
<tr>
<td>signal:eucl</td>
<td>$\sqrt{\sum_{1}^{N} (a_i - b_i)^2}$</td>
<td>The euclidean difference between received signal strength vectors.</td>
</tr>
<tr>
<td>signal:exp</td>
<td>$\frac{\sum_{1}^{N} e^{</td>
<td>a_i - b_i</td>
</tr>
<tr>
<td>signalexp:diff:eucl</td>
<td>$\sqrt{\sum_{2}^{N} (e^{(a_i-a_{i-1})} - e^{(b_i-b_{i-1})})^2}$</td>
<td>The euclidean difference between exponential signal strength deltas.</td>
</tr>
</tbody>
</table>
Base-line Testing

- Same lab, moved 10 meters apart and 2 months after training
- Same machine layout, attackers had line of sight on victims
- Testing as a function of classifiers window size
- Only the 1m away attacker is successful with windows > 3 seconds (success rate ~ 60%)
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Feature Definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>signal:abs</td>
<td>$P_N</td>
<td>a_i - b_i</td>
</tr>
<tr>
<td>signal:eucl</td>
<td>$\sqrt{\sum N (a_i - b_i)^2}$</td>
<td>The euclidean difference between received signal strength vectors.</td>
</tr>
<tr>
<td>signal:exp</td>
<td>$P_N e^{</td>
<td>a_i - b_i</td>
</tr>
<tr>
<td>signal:exp</td>
<td>$\sqrt{\sum (e^{a_i - a_{i-1}} - e^{b_i - b_{i-1}})^2}$</td>
<td>The euclidean difference between exponential signal strength deltas.</td>
</tr>
</tbody>
</table>

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 $b_i$, $0 \leq i \leq N$.

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

Figure 5: The effect of handwaving on the ability of the attacker to authenticate after 10 seconds when he is 1m, 0.5m, 0.2m and 0.05m from the friendly devices.

4.4 The Limits of Handwaving
In the previous section, we showed that our handwaving technique works well in the case where the distance between the attacker and the friendly devices is 1m. To test whether handwaving continues to work when the attacker is even closer, we collected additional traces in a new location with the attacker at distances of 1m, 0.5m, 0.2m and 0.05m from the friendly devices.

Figure 5 shows the false positive rate after 10 seconds for the four attacker distances. The results show that at 1m and 0.5m handwaving works well, with not a single pairing attempt by the attacker being successful. At 0.2m, the attacker is able to pair with one of the co-located devices in 2% of his attempts and at 0.05m, 40% of his attempts are successful. We believe that this is possibly because the attacker is so close that the handwaving effect the signals received by both the attacker and the friendly devices in similar ways.

4.5 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 6 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.
Hand-waving

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

- **signal:abs**
  \[ |a_i - b_i| \]
  The mean absolute difference between received signal strength measurements.

- **signal:eucl**
  \[ \sqrt{\sum_{N} (a_i - b_i)^2} \]
  The Euclidean difference between received signal strength vectors.

- **signal:exp**
  \[ e|a_i - b_i| \]
  The mean exponential of the difference between signal strength measurements.

- **signalexp:diff:eucl**
  \[ \sqrt{\sum_{N} (e(a_i - a_{i-1}) - e(b_i - b_{i-1}))^2} \]
  The Euclidean difference between exponential signal strength deltas.

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

Figure 5: The effect of handwaving on the ability of the attacker to authenticate after 10 seconds when he is 1m, 0.5m, 0.2m and 0.05m from the friendly devices.

4.4 The Limits of Handwaving
In the previous section, we showed that our handwaving technique works well in the case where the distance between the attacker and the friendly devices is 1m. To test whether handwaving continues to work when the attacker is even closer, we collected additional traces in a new location with the attacker at distances of 1m, 0.5m, 0.2m and 0.05m from the friendly devices.

Figure 5 shows the false positive rate after 10 seconds for the four attacker distances. The results show that at 1m and 0.5m handwaving works well, with not a single pairing attempt by the attacker being successful. At 0.2m, the attacker is able to pair with one of the co-located devices in 2% of his attempts and at 0.05m, 40% of his attempts are successful. We believe that this is possibly because the attacker is so close that the handwaving effect on the signals received by both the attacker and the friendly devices in similar ways.

4.5 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 6 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
Effects of Margin Threshold

![Graph showing the effects of margin threshold on false positive and false negative rates.]

- False Positive Rate
- False Negative Rate

**False Positive Rate** and **False Negative Rate** as a function of margin threshold. The graph illustrates how the false positive and false negative rates change with varying margin thresholds. Notably, as the margin threshold increases, the false positive rate decreases significantly, while the false negative rate remains relatively stable. The data suggests that selecting an appropriate margin threshold can effectively balance the trade-off between these two metrics, potentially improving the overall performance of the system.
Effects of Obstacles, Orientation

<table>
<thead>
<tr>
<th>Obstruction</th>
<th>False Positive Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>None (1m)</td>
<td>0.81</td>
</tr>
<tr>
<td>Drywall (10cm)</td>
<td>1.00</td>
</tr>
<tr>
<td>Human (1m)</td>
<td>0.12</td>
</tr>
<tr>
<td>Concrete Wall (30cm)</td>
<td>0.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Orientation</th>
<th>False Positive Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opposite (5cm)</td>
<td>0.0</td>
</tr>
<tr>
<td>Adjacent (30cm)</td>
<td>0.0</td>
</tr>
<tr>
<td>Stacked (2cm)</td>
<td>0.14</td>
</tr>
</tbody>
</table>
Devices at 1m

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

<table>
<thead>
<tr>
<th>Material</th>
<th>False Positive Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>None (1m)</td>
<td>0.0</td>
</tr>
<tr>
<td>Drywall (10cm)</td>
<td>1.00</td>
</tr>
<tr>
<td>Human (1m)</td>
<td>0.12</td>
</tr>
<tr>
<td>Concrete Wall (30cm)</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 3: The effect of device orientation on the ability of the attacker to authenticate with a device. The distances in parentheses are between the friendly devices’ wireless interfaces in each configuration.

<table>
<thead>
<tr>
<th>Orientation</th>
<th>False Positive Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opposite (5cm)</td>
<td>0.0</td>
</tr>
<tr>
<td>Adjacent (30cm)</td>
<td>0.0</td>
</tr>
<tr>
<td>Stacked (2cm)</td>
<td>0.14</td>
</tr>
</tbody>
</table>

Figure 7: Co-located devices are up to 1m apart. The attacker is 3m, 5m and 10m away.

4.9 Sophisticated Attacks

A powerful attacker may 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 in which 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 8 shows that attackers located 1 meter and 5 meters away can successfully authenticate in 45% and 15% 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 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
Sophisticated Attacker

- Attacker has access to signal strength values from target device from a few hours in the past
- During attack, when a packet from a known source is seen, signal strength from recorded target values is inserted
- For new sources, attacker can ignore or leave signal strength unaltered
- Testing shows later approach is superior for attacker
Sophisticated Attacker, continued

We tested whether hand waving will prevent this attack. In a worst-case scenario, an attacker could also be powerful, as the figure shows the false positive rate of authenticating with the target device in this case. Figure 9 shows the false positive rate of authenticating with the oracle attacker. The figure shows that the case. Figure 9 shows the false positive rate of authenticating with the oracle attacker.

In this section, we report the entropy of the radio signature. Recall that a radio signature consists of an ordered sequence of packet hashes and associated discrete RSSI values. Note that the numbers reported are an upper bound of heterogeneous hardware on the accuracy of Amigo, we conducted an experiment using different types of 802.11g capable wireless interfaces: two ZyXel cards with different chipsets, but the attacker has the same hardware. Therefore, the signal strength values as received by various chipsets [Haeberlen et al., 2004] need to be normalized before they can be used.

In order to make Amigo work with common WiFi interfaces, WiFi cards need to be normalized before they can be used. Therefore, the signal strength values as reported by the two new cards and the mapping was computed by minimizing the difference between sequences of signal strength and RSSI values as reported by various chipsets. Before starting experiments, we computed a linear mapping between signal strength values as reported by the Orinoco card. The mapping was computed by minimizing the difference between sequences of packet hashes and associated discrete RSSI values. So far, we have evaluated Amigo using the same WiFi cards. In order to test the effect of heterogeneous hardware on the accuracy of Amigo, we conducted an experiment using different chipsets, but the attacker has the same hardware.

Figure 8 shows the false positive rate for a simulated attacker who can accurately predict what packets the target device receives. Also, for our experiments, we assume that such an attacker has injected every packet into the environment, and that he knows exactly where the pairing takes place will have to guess the location where the pairing takes place will have to guess the location where the pairing takes place will have to guess the location where the pairing takes place will have to guess the location where the pairing takes place will have to guess. However, blindly guessing a sequence of packets in a radio signature and the RSSI value will result in very large entropy. Therefore, the attacker is equipped with an oracle, that allows the attacker to sample from the distribution of co-located devices and then uses that signal strength to impersonate one of the devices.

In this section, we report the entropy of the radio signature. Recall that a radio signature consists of an ordered sequence of packet hashes and associated discrete RSSI values. Note that the numbers reported are an upper bound of heterogeneous hardware on the accuracy of Amigo, we conducted an experiment using different types of 802.11g capable wireless interfaces: two ZyXel cards with different chipsets, but the attacker has the same hardware. Therefore, the signal strength values as received by various chipsets [Haeberlen et al., 2004] need to be normalized before they can be used.

In order to make Amigo work with common WiFi interfaces, WiFi cards need to be normalized before they can be used. Therefore, the signal strength values as reported by the two new cards and the mapping was computed by minimizing the difference between sequences of signal strength and RSSI values. So far, we have evaluated Amigo using the same WiFi cards. In order to test the effect of heterogeneous hardware on the accuracy of Amigo, we conducted an experiment using different chipsets, but the attacker has the same hardware.

Figure 8 shows the false positive rate for a simulated attacker who can accurately predict what packets the target device receives. Also, for our experiments, we assume that such an attacker has injected every packet into the environment, and that he knows exactly where the pairing takes place will have to guess the location where the pairing takes place will have to guess. However, blindly guessing a sequence of packets in a radio signature and the RSSI value will result in very large entropy. Therefore, the attacker is equipped with an oracle, that allows the attacker to sample from the distribution of co-located devices and then uses that signal strength to impersonate one of the devices.

In this section, we report the entropy of the radio signature. Recall that a radio signature consists of an ordered sequence of packet hashes and associated discrete RSSI values. Note that the numbers reported are an upper bound of heterogeneous hardware on the accuracy of Amigo, we conducted an experiment using different types of 802.11g capable wireless interfaces: two ZyXel cards with different chipsets, but the attacker has the same hardware. Therefore, the signal strength values as received by various chipsets [Haeberlen et al., 2004] need to be normalized before they can be used.
Oracle attacker w/ hand waving

Figure 8: False positive rate for a simulated attacker who authenticates with the target device in this case. Figure 9 shows the false positive rate of authentication attempts with the oracle attacker. The figure shows that the false positive rate falls below 5% immediately, and reaches 0% after the duration of the experiment.

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 some additional knowledge of the radio domain.

An attacker that is not physically close to the actual location where the pairing takes place will have to guess the packet hashes will result in very large entropy. Therefore, we assume that the attacker is located somewhere nearby packet hashes will result in very large entropy. Therefore, we assume that the attacker is located somewhere nearby.

Recall that a radio signature consists of an ordered sequence of packet hashes and associated discrete RSSI values. Note that the numbers reported are an upper bound on the entropy, meaning that the actual entropy may be lower if an attacker doesn’t simply guess values, but uses some additional knowledge of the radio domain.

In this section, we report the entropy of the radio signature as perceived by the target device over the duration of the experiment.

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 some additional knowledge of the radio domain.

Therefore, the signal strength values as received by various chipsets [Haeberlen et al., 2004] need to be normalized before they can be used for authentication. This normalization typically takes the form of a linear transformation between signal strength values as reported by the two new cards and the Orinoco card. The mapping was computed by minimizing differences that we used for training the classifier. However, WiFi cards need to be normalized before they can be used effectively, we tested different card combinations, collected simultaneously at a single USB dongle played the role of an attacker, trying to authenticate to the first USB dongle. E

So far, we have evaluated Amigo using the same WiFi cards different types of 802.11g capable wireless interfaces: two ZD1211-based 802.11b/g USB dongles and an Atheros-based 802.11b/g PCMCIA card. Before starting experiments, we computed a linear mapping between signal strengths and provide RSSI values in different ranges.

4.10 Difference between sequences of signal strength distribution of packets. However, blindly guessing a sequence of sequence of packets in a radio signature and the RSSI value allows the attacker to sample from the distribution of co-located devices and then uses that signal strength to impersonate one of the devices.

In the current experiments, we tested our classifier with the new traces. For each packet, we applied the linear transformations on the signal strength values as reported by the two new cards and the Orinoco card. The mapping was computed by minimizing differences that we used for training the classifier. However, WiFi cards need to be normalized before they can be used effectively, we tested different card combinations, collected simultaneously at a single USB dongle played the role of an attacker, trying to authenticate to the first USB dongle. E

Figures 10(a) and 10(b) plot the false positive and false negative rates as a function of time the user needs to wait before pairing the devices. E
Figure 10: A simple normalization of signal strength makes Amigo work well with heterogeneous WiFi cards.

To calculate the number of bits of entropy per second in a radio signature, we first calculate the number of bits of entropy there are per packet and then multiply it by the average number of packets per second in the signature. In our experiments, packets were received with a total of 41 different RSSI values, which results in 5.36 bits of entropy per packet. Since, on average, laptops received 160.15 packets per second, the entropy of the radio signatures is thus 858 bits per second. However, in practice, the attacker might try to guess a rough range of RSSI values per packet if he is located in a vicinity. If the attacker is able to guess the RSSI range down to 10 different RSSI values or ranges, the entropy becomes 532 bits per second. Moreover, if the attacker reduces the range of the RSSI values down to 5 possibilities, the entropy is still 372 bits per second.

Although we've established only an upper-bound on the entropy of the signatures used by Amigo, we believe that the true entropy compares favorably with that of keys used in widely-deployed authentication systems such as Bluetooth, with approximately 13 bits of entropy using a 4-digit PIN.

5 Related Work

SWAP-CA [SWAP-CA, 1998] 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 [Cheung, 2005, Shaked and Wool, 2005]. LoKey [Nicholson et al., 2006] 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 [Holmquist et al., 2001] 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 [Mayrhofer and Gellersen, 2007]. Both of these techniques use the accelerometer readings as the basis of the authentication. In Shake Them Up! [Castelluccia and Mutaf, 2005], 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 [Stajano and Anderson, 1999], infrared
Entropy

- Upper bound: 858 bits/second
  - If RSSI limited to 10 values: 532 bits/second
  - If RSSI limited to 5 values: 372 bits/second
- Bluetooth
  - 13 bits for 4-digit pin
Related work

- Bluetooth
  - Exchange 4 digit pin
- Smart-It
  - Shaking two things together and using accelerometer readings
- Physical channel
  - Requires physical hardware, sensitive to powerful antennas
- Near Field Communication
Conclusion

- Provides authentication of secure communication channel for mobile devices
- Amigo detects attackers as close as 3 meters given 5 second window
- If willing to wave hand, limits attackers to 1m or closer
- Requires no extra hardware
Thank you

✤ Comments?

✤ Questions?

✤ Joe Buck can be contacted at buck@soe.ucsc.edu