Purdue University

RF-Draw Security

Krutarth Rao and Anthony Geockner

November 19, 2017

Overview

RF-Draw uses a mesh network powered by Zigbee compatible radio frequency devices. Every “draw” device broadcasts the input received from it’s touch screen sensor to every other device in the network. The collective input from the draw devices constructs the **diagram**. Devices in the network can be display devices or draw devices.

Draw device: Broadcasts all touch screen device inputs to the network while listening for messages from other display devices.

Display device: Is a listen only device. Listens to all draw devices in the mesh network. *(One display device gets connected to the large display of the classroom/conference room)*

For the security aspects of the network, we consider two adversary models:

Active Adversary:

- Attempts to influence the **diagram** by introduced an unauthorized device into the mesh.
- Flood the mesh with arbitrary messages that and cause the network DoS.

Silent Adversary:

- Listen to network to collect information about the diagram being created by sniffing the packets being exchanged in the mesh.

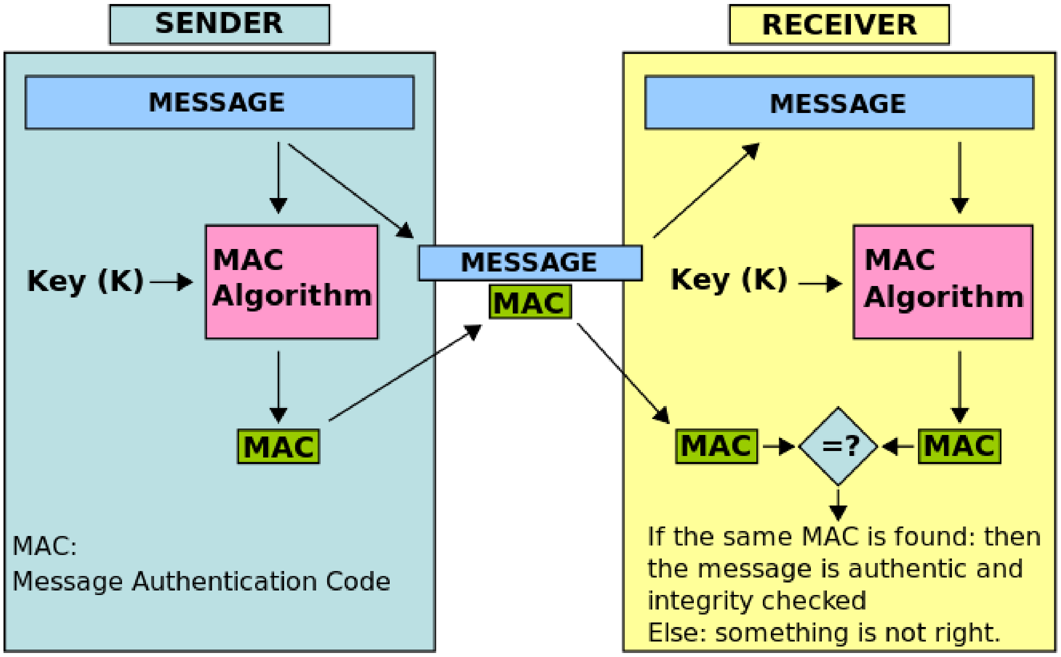
Solution and Technical Details

For our implementation, we consider an active adversary attempting to influence that diagram as the relevant threat in the primary usage of the system (classroom). The issue can be mitigated by using Message Authentication Codes (MAC).

MACs are cryptographically secure primitives that can authenticate the messages that come from parties in a network that have agreed upon a key prior to their communication. Our implementation is as follows:

- Presenter announces a *pin* the class will use in their devices. And all users with the device enter the key before they begin using the network.
- Each device computes a SHA256 hash of the entered pin and stores it as the session key in memory. (*The hash allows for 256-bit keys that provide higher security than shorter keys*)
- Each packet prepared to be sent to the mesh is concatenated with its MAC.
- Each device checks for a valid MAC when it receives a packet and drops the packet if the check fails.

The protocol above can be represented as follows:


SENDER
MESSAGE
Key (K)
MAC Algorithm
MAC
MAC:
Message Authentication Code
MESSAGE
MAC
RECEIVER
MESSAGE
Key (K)
MAC Algorithm
MAC
=?
MAC
If the same MAC is found: then the message is authentic and integrity checked
Else: something is not right.

MAC Algorithm

The above section abstracts the MAC algorithm for simplicity. For a truly secure MAC, we consider the following facts about weak MAC implementations:

- Chosen hash function is not truly randomized. Introducing a fundamental flaw in HMAC. (check?)
- Incorrect masking of the key in the hash function call can expose length extension attacks.

Our construction uses the HMAC algorithm. Using the key k generated by the SHA256 hash above, we define the hash of a message m as follows:

$$H_k(m) = F(k||F(k||m))$$

Where $||$ is the concatenation operation. And F represents the chosen hash function for the session. (can using two different hash functions improve security?)

The proposal does not go into the details of the *ipad* and *opad* used in the hash computation so the focus remains on the system level details of the message authentication. However, more details can be found in[1] and[2].

[1] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. "Keying hash functions for message authentication". In: *Crypto*. Vol. 96. Springer. 1996, pp. 1–15.

[2] Harris E Michail et al. "Efficient implementation of the keyed-hash message authentication

We finally show that HMAC is a secure MAC scheme by considering the following properties:

- CPA
- Existential forgery

Protocol Termination

Once a device if powered off, the session key is effectively lost as the key is never stored outside main memory and main memory remains non-persistent in the chosen devices.

References

Bellare, Mihir, Ran Canetti, and Hugo Krawczyk. “Keying hash functions for message authentication”. In: *Crypto*. Vol. 96. Springer. 1996, pp. 1–15.

Michail, Harris E et al. “Efficient implementation of the keyed-hash message authentication code (HMAC) using the SHA-1 hash function”. In: *Electronics, Circuits and Systems, 2004. ICECS 2004. Proceedings of the 2004 11th IEEE International Conference on.* IEEE. 2004, pp. 567–570.

code (HMAC) using the SHA-1 hash function”. In: *Electronics, Circuits and Systems, 2004. ICECS 2004. Proceedings of the 2004 11th IEEE International Conference on.* IEEE. 2004, pp. 567–570.