Attacks that our team developed:

- Timing attacks to break timing-reliant RNG
- ROP chains to exploit buffer overflows
- Replay attacks with deployment-wide secrets
- Message forging with secrets leaked in features
- Nonexistent or improperly placed delays allowing PIN brute force
- Misuse of strncmp for memory comparisons allowing comparisons to be bypassed
- Custom key exchange protocol allowing decryption of all traffic and forging of unlock requests
- Lack of authentication or encryption allowing enabling of features

A couple of teams use the internal temperature sensor readings as part of their RNG. This sensor is read using the onboard ADCs, which rely on two reference pins VDDA and GNDA. By disconnecting GNDA from the board and shorting it with VDDA, we are able to lower the ADC reading by 0x200. This lowers the required temperature to saturate the ADC reading from around 145°C to 117°C. One team uses a Von Neumann extractor to generate random numbers, so saturating the ADC reading alone would not work. Instead, we can toggle a relay connecting VDDA and GNDA at the sample rate of the ADC, ensuring that readings will alternate between saturated and non-saturated, hopefully producing a predictable value. In the end, we didn’t have the time to carry out the attack.

A variety of different RNG sources like we used in our design would largely mitigate this sort of attack.