IITM

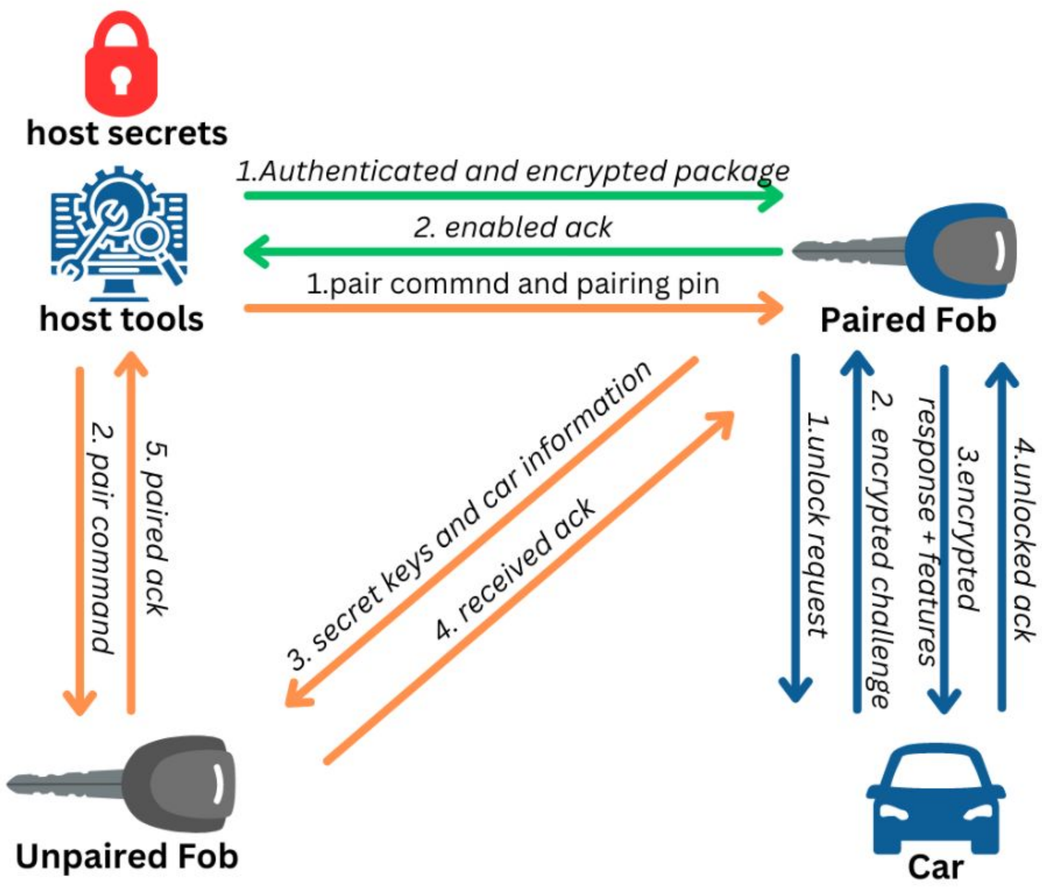
Indian Institute of Technology, Madras

**Arun Krishna, Divya Prajapati, Antonson, Priyanka, Shree Vishnu,
Reetwik Das, Ritwik Badola, Mahesh
Advised by: Dr. Chester Rebeiro, PhD**

Design Overview:

Our entire design philosophy revolved around the use of symmetric encryption because of the memory and computation power limitations. We used lightweight cipher ASCON with AEAD to provide confidentiality, integrity, and authenticity of the messages. Designs without them are prone to Man-in-the-middle attacks.

- **Unlock Process:** Randomly generated Challenge-Response (based on ASCON encryption/decryption) is used to authenticate the car and the fob. Generated Challenge is valid only within a specific time. This prevents unauthorized access, replay, and roll jam attacks
- **Pairing Process:** The car-unlock and feature-enable keys are sent to the unpaired fob by the paired fob only when the pairing pin matches.
- **Enable Feature:** Feature is packaged, and encrypted by the manufacturer. The fob decrypts and authenticates the package using its pre-shared key.

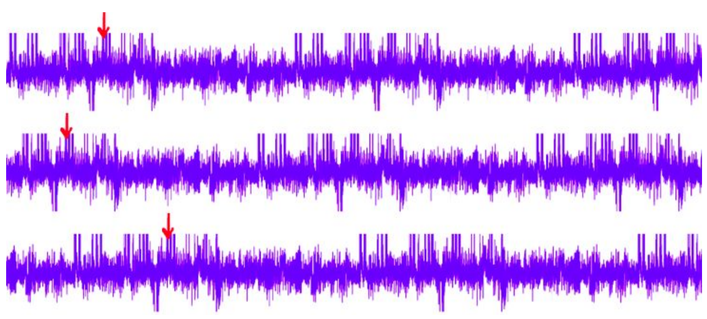


Defensive Highlight:

Fault & Side Channel Protection

While the protocol seems secure, it can be broken by Side Channel & Fault-Injection attacks. While the later attacks can't be prevented, we add the following countermeasures that render the attacks more complex.

- **Adding random time delays:** Random delays in the execution introduce desynchronization when collecting traces, increasing the complexity of side-channel attacks. Because of random delays, the target point in time is constantly changing its position. Thus the attacker has to inject faults numerous times until the target & fault coincide.



Desynchronized traces

- **Clock Glitching - Fault Injection Protection:** The external oscillator on the evaluation board is prone to tampering. Fault injection through clock glitching is prevented in our design because our design utilizes precision internal oscillator which cannot be tampered externally.
- **ASCON:** Use of side-channel resistant cipher.

The design can be made further resistant against fault-injection attacks by doing critical calculations repeatedly and checking for any inconsistencies.

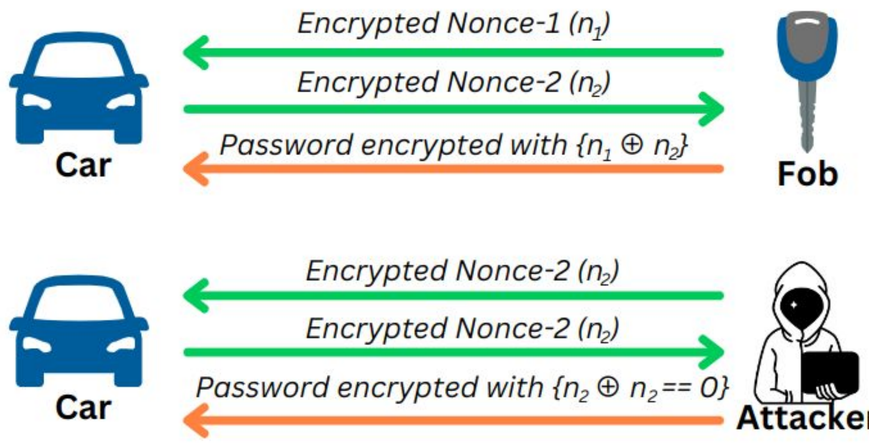
Design Improvements:

During the pairing process, if the pairing pin is correct, the paired fob openly shares the car key secrets with the unpaired fob without any encryption. Thus the pairing pin is a single point of failure in the current design. The car key secrets must be encrypted when sending them to the unpaired fob to avoid snooping.

Symmetric encryption with a fixed key cannot be used here, because the attacker can obtain the fixed key from Car 0. Instead, a session key for the pairing process must be created on the fly using authenticated secure key exchange protocol for the encrypted sharing of car secret keys. Since the private keys are not known to other players, the confidentiality can be ensured.

Offensive Highlight:

To prevent replay attacks, one of the teams designed their protocol to generate session keys for each unlocking session. To generate the session key, the fob generates an encrypted nonce n1 and shares it with the car. Following that, the car also generates an encrypted nonce n2 and shares it with the fob. Both the car and fob decrypt the received nonces, XORs them (n1 ^ n2) and use the result as the session key.



By forcing the car to choose the same nonce as the one fob sends, the attacker can force the session key to be 0. Since the message and key are known, the attacker can unlock the car. Their design is vulnerable because the generated nonce was not completely random.

Their nonces were generated from a constant SEED which was stored in EEPROM. Although it gets updated every time a random number is generated, this method can be broken because the same nonce is observed after re-flashing the firmware, thus making the entire design vulnerable.

To prevent this the RNG must be strong. Since TM4C123 doesn't have hardware RNG, the seed for the RNG must be from multiple other sources such as:

- Initial values for SRAM at the startup (SRAM must be de-initialized in the linker file for this to work - one of the teams forgot to do this)
- Debiased noise from internal temperature sensors of ADC
- Unknown fixed salt generated during the firmware build process.

The entropy for the RNG can be further improved by also utilizing system/CPU ticks.

References

ASCON Implementation:
https://ascon.iaik.tugraz.at/index.html