Security evaluation of countermeasures against physical attacks inserted at compilation time

Brice Colombier, Pierre-Alain Moëllic
CEA - DRT/DPACA, Laboratoire SAS
Centre de Microélectronique de Provence, Site Georges Charpak,
880 Avenue de Mimet, 13120 Gardanne, France

Abstract

With the advent of multiple forms of side-channel and fault attacks in the last decades, it has become standard practice to integrate some countermeasures in embedded code to thwart these attacks. Manual insertion of these countermeasures requires expertise from the designer. Moreover, the compilation flow must be adapted as well to cope with these non-functional features and not remove them.

Instead, the approach that we follow is to modify the compiler directly to protect code that has been previously annotated in a simple way. This allows a non-expert designer to protect sensitive variables and instructions easily. The two typical use cases to protect that we consider here are a VerifyPIN program and AES encryption algorithm. The former is protected against fault attacks while the latter is protected against fault attacks as well as side-channel attacks.

We evaluate the security of these protections in different scenarios. The first one is a low-cost context where Correlation Power Analysis and clock glitches were performed using the ChipWhisperer platform. The second scenario is more high-end, focusing on complex side channel attacks against a masked AES implementation using deep learning techniques and fault attacks with a laser source.

We evaluate the overall resistance of the countermeasures against this range of attacks and provide a comprehensive presentation of the cost/efficiency trade-off involved, both from a defender and attacker point of view. This paves the way to developer-friendly tools that would allow them to easily protect embedded code.