TUTORIALS 7324

CRYPTOGRAPHY: PLANNING FOR THREATS AND COUNTERMEASURES

By: Zia Sardar

Abstract: This tutorial is part of a series that is designed to provide a quick study guide in cryptography for a product development engineer. Each segment takes an engineering rather than theoretical approach on the topic. In this installment, you'll learn about the different types of threats cryptographic systems face, learn how to plan for threats, and see what types of countermeasures are available. A similar version of this tutorial originally appeared on Electronic Design on June 19, 2020.

Threats, Countermeasures, and Security Planning

When we consider connected systems, it's important to recognize that such systems do not only mean those connected to the internet. A connected system can include a pulse oximeter that is connected to a patient in a hospital environment, or a printer cartridge that is connected to a printer. All of our connected systems face constant security threats from various sources. This means that the current plethora of IoT (Internet of Things) devices like thermostats and refrigerators are also susceptible to hacking. Let's look at a few of these threats, learn how to protect your devices, and see what kind of planning you need to do.

Threats

Developers today face threats to systems as well as to security ICs. Threats to systems have been well covered by other sources, so we will only focus on threats to security ICs. A security IC can be attacked by one or more of the following methods:

- Side-channel attacks, such as a glitch attack (active) and differential power analysis (passive).
- Invasive attacks, such as decapping and micro-probing to find open ports and traces that can be exploited.
- Line snooping, such as a man-in-the-middle attack.
- Memory array tampering, such as a cold boot attack.

Most of the time, side-channel attacks are noninvasive attacks, i.e., they do not destroy the IC. Decapping and micro-probing, which physically investigate various features of the IC, on the other hand, are invasive attacks that can destroy the IC.

We are not going to go into too much detail about how these attacks are carried out, but we will show a couple of simple examples.

Active Side-Channel Attack: Glitch Attack

A side channel includes any information that is available as a side effect of the physical implementation of hardware. This may include power consumption, injection of faults, etc. **Figure 1** shows a type of side-channel attack using clock glitches. This is an example of a noninvasive attack.

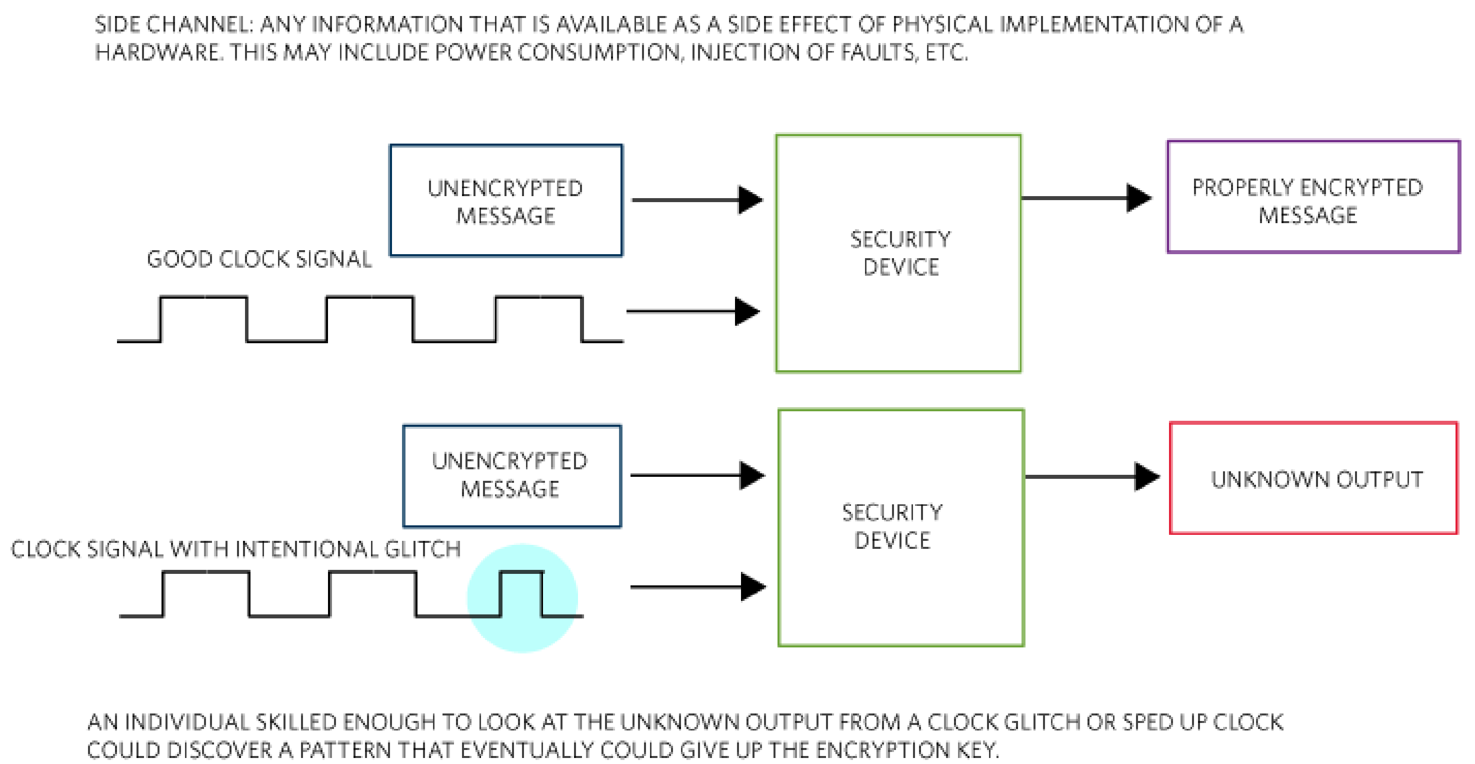

Figure 1. Active side-channel attack is an example of a noninvasive attack.

An individual skilled enough to look at the unknown output from a clock glitch or sped-up clock could discover a pattern that could eventually reveal an encryption key.

Decapsulation

Decapsulation, also known as de-capping, involves soaking the plastic package that encapsulates the silicon die in fuming nitric acid to melt away the package (**Figure 2**). Normally, before that is done, the lead frame that holds the semiconductor die is secured on a frame. This is considered an invasive attack.

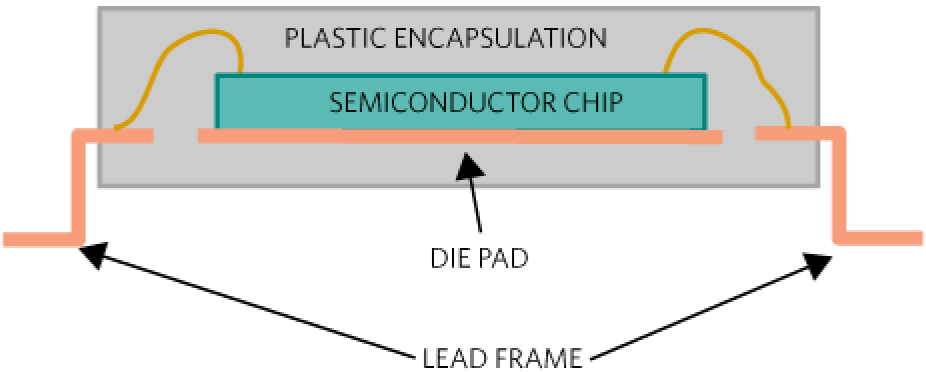

Figure 2. Semiconductor packages are vulnerable to invasive attacks.

Semiconductor Package

Once the package is melted away, the die gets exposed, providing the hacker an opportunity to directly probe all the available pads, including pads that the manufacturer used for internal setup (**Figure 3**). They can also polish away the top protective glass and can access the internal interconnects of the device. Using this direct method, the hacker will try to gain access to the device's secrets.

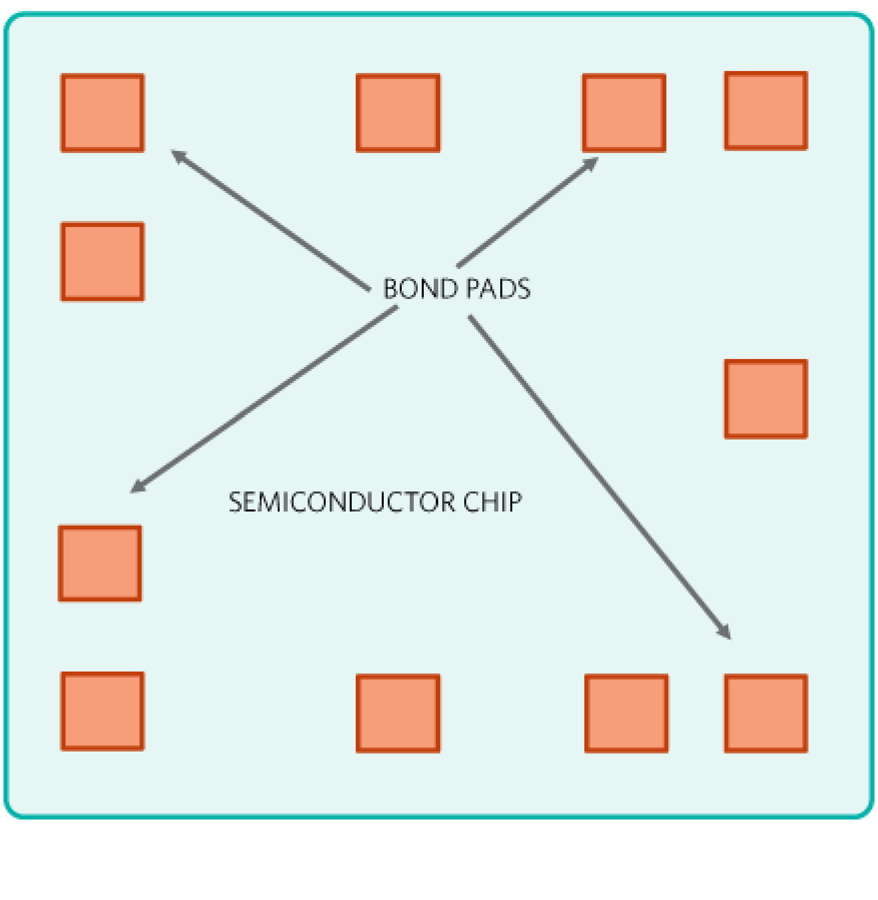

Figure 3. Hackers can directly probe available pads on a chip (top view).

Countermeasures

To prevent people with malicious intent from breaking into a secure device, the device must be designed with features that not only provide security but also protect the device from attacks. Maxim's secure devices have robust countermeasures to protect against all these attacks. Here are some of the implemented features:

- Patented physically unclonable function (PUF) technology to secure device data.
- Actively monitored die shield that detects and reacts to intrusion attempts.
- Cryptographic protection of all stored data from discovery.

Security Planning

Depending on the application need, the user must decide which cryptographic features are appropriate to deploy. **Table 1** shares some examples of application needs and the resultant measures that need to be applied.

Table 1. Security Planning Based on Application Security Needs

	Authenticity	Confidentiality	Integrity
Against Counterfeiting	X		
Against Eavesdropping		X	
Against Malware Injection	X		X
Against Calibration Data Change	X		X

For example, if someone is trying to prevent a medical surgical tool from being counterfeited, they must ensure that every time a tool is connected to the host controller (**Figure 4**), the tool's authenticity is checked. It will also need protection against any malware from being installed in the tool, which could potentially harm the patient. The need to protect any calibration data that was stored is paramount as well. But as the possibility of snooping between the tool and the host controller is next to impossible due to closed system connectivity, this system will not need protection against eavesdropping. Thus, in this case, the system designer needs to plan for all the protections under the "Authenticity" column but can skip unnecessary protection listed under the "Confidentiality" column.

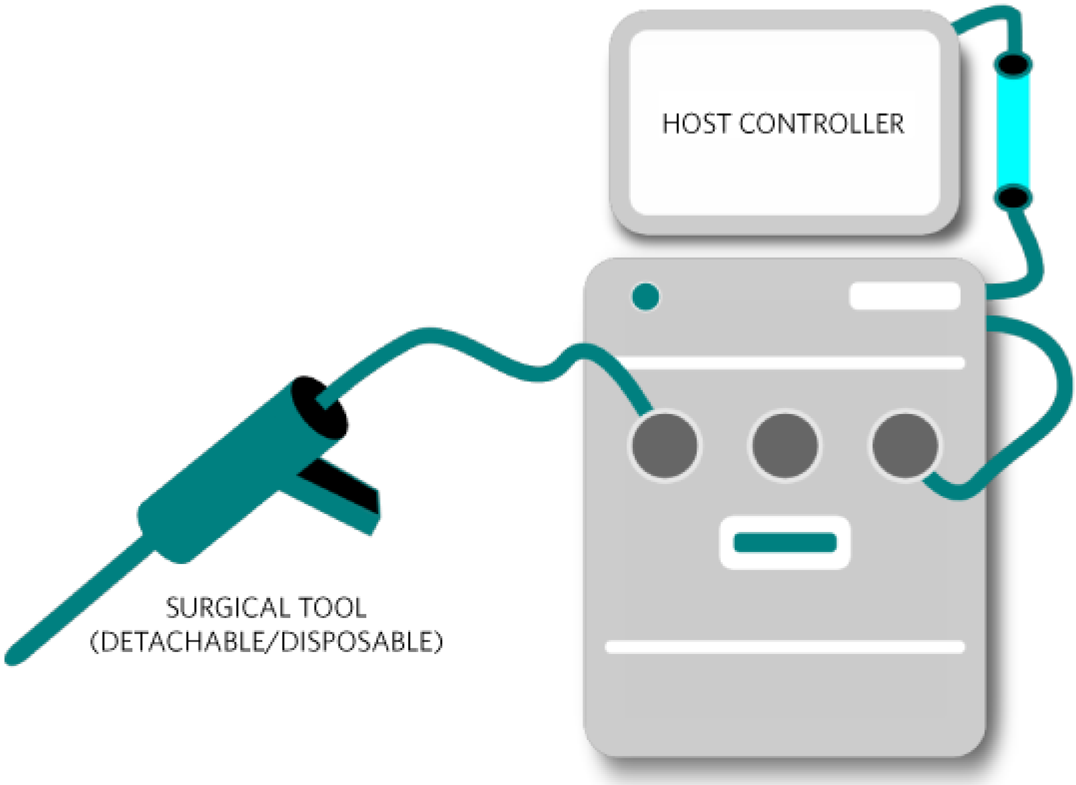

Figure 4. Security planning should include counterfeit prevention for medical devices like surgical tools.

Example System—Security Planning

Thus, in conclusion, threats to ICs are ever present and from many sources. A system designer needs to be aware of the types of threats and plan accordingly. This article gave you some simple insights into how that can be achieved. But if you are a busy product developer with a tight deadline, do you really have the time to be an expert in this field as well as complete your design on time? Probably not. That's where the use of a secure authenticator designed and built by industry experts may be a good option. The next installment in this series of cryptography tutorials will discuss secure authenticators.

Related Parts		
DS2476	DeepCover Secure Coprocessor	Free Sample
DS2477	DeepCover Secure SHA-3 Coprocessor with ChipDNA PUF Protection	Free Sample
DS28E16	1-Wire Secure SHA-3 Authenticator	Free Sample
DS28C16	I²C Low-Voltage SHA-3 Authenticator	Free Sample
DS28E36	DeepCover Secure Authenticator	Free Sample

Related Parts		
DS28C36	DeepCover Secure Authenticator	Free Sample
DS28E38	DeepCover® Secure ECDSA Authenticator with ChipDNA PUF Protection	Free Sample
DS28E39	DeepCover Secure ECDSA Bidirectional Authenticator with ChipDNA PUF Protection	Free Sample
DS28C39	DeepCover Secure ECDSA Bidirectional Authenticator with ChipDNA PUF Protection	Free Sample
DS28E50	DeepCover Secure SHA-3 Authenticator with ChipDNA PUF Protection	Free Sample
DS28C50	DeepCover I²C Secure SHA-3 Authenticator with ChipDNA PUF Protection	Free Sample
DS28E83	DeepCover Radiation Resistant 1-Wire Secure Authenticator	Free Sample
DS28E84	DeepCover Radiation Resistant, High-Capacity 1-Wire Secure Authenticator	Free Sample
MAX66240	DeepCover Secure Authenticator with ISO 15693, SHA-256, and 4Kb User EEPROM	Free Sample
MAX66242	DeepCover Secure Authenticator with ISO 15693, I²C, SHA-256, and 4Kb User EEPROM	Free Sample
MAXQ1061	DeepCover Cryptographic Controller for Embedded Devices	Free Sample
MAXQ1062	DeepCover Cryptographic Controller for Embedded Devices	Free Sample

Next Steps	
EE-Mail	Subscribe to EE-Mail and receive automatic notice of new documents in your areas of interest.

APP 7324: 20 Jul, 2020
TUTORIALS 7324, AN7324, AN 7324, APP7324, Appnote7324, Appnote 7324