Hardware Trojans in Wireless Cryptographic ICs

Yier Jin and Yiorgos Makris
Yale University

Editor's note:
This article studies the problem of hardware Trojans in wireless cryptographic ICs. The objective is to design Trojans to leak secret information through the wireless channel. The authors investigate challenges related to detection for such Trojans and propose using statistical analysis of the side-channel signals to help detect them.
—Mohammad Tehranipoor, University of Connecticut

The problem of maliciously intended modifications, commonly known as *hardware Trojans*, in manufactured ICs has recently garnered interest not only in academia but also in governmental agencies and industry.[1] Partly because of design outsourcing and migration of fabrication foundries to low-cost areas across the globe, and partly because of increased reliance on external hardware IP and EDA software from various vendors, the IC supply chain is now considered far more vulnerable to such malicious modifications than ever before. In essence, the fundamental concern is that hardware Trojan-infested chips might be capable of additional functionality that the designer, vendor, and customer are unaware of and which the perpetrator can exploit after chip deployment. Depending on the field of application, the consequences of such attacks can range from minor inconvenience to major catastrophes.

Rad et al. recently introduced a comprehensive taxonomy of hardware Trojans based on their physical, activation, and action characteristics, discussing the concepts of type, trigger, and payload.[2] *Type* refers to the form of attack and the way it is physically staged; hardware Trojans can aim either at a chip's logic functionality or parametric functionality and can be implemented in either a localized or distributed fashion. *Trigger* refers to the mechanism that enables the maliciously added functionality; hardware Trojans might be always active or might rely on specific events (e.g., input sequences and elapsed time) to be activated. *Payload* refers to the actual impact of the maliciously added functionality; hardware Trojans might distort the results, incur denial of service, or even physically incapacitate the chip.

Among the possible approaches to address this problem, traditional production testing falls short because it is designed to cost-effectively detect manufacturing defects but is not geared toward uncovering such malicious hardware modifications. Destructive reverse-engineering might be an option, but it becomes increasingly expensive and difficult as chip complexity increases. Furthermore, as the name indicates, it can only be used on a small sample of chips and provides no guarantees that the remaining chips are Trojan free. In identifying the need for trusted ICs, several researchers have already rushed to develop methodologies that are particularly tuned to the detection of hardware Trojans. A close look at the various Trojan detection schemes that have been proposed reveals two main directions: enhanced functional testing and side-channel fingerprint generation and checking. Along the first direction, Wolff et al. hypothesized that attackers will choose rarely occurring events as triggers and proposed the idea of identifying such events and enhancing the production test set accordingly.[3] Along the second direction, the work described by Agrawal et al.[4] is a significant milestone because it first demonstrated the utility of statistical analysis toward constructing effective power-based fingerprints to distinguish genuine from Trojan-infested ICs. Alternatively, we proposed the use of path delay fingerprints in earlier work.[5] Rad et al. proposed a similar strategy based on anomalies introduced by a hardware Trojan to the currents measurable at the power ports of a chip.[6] This general method, which relies on dividing the power supply

0740-7475/10/$26.00 © 2010 IEEE Copublished by the IEEE CS and the IEEE CASS

network into smaller power grids, was further enhanced by employing various measurement calibration techniques to reduce the adverse effects of process and test environment variations.[2]

In contrast to these methods, which target digital circuits, we focus here on wireless cryptographic ICs. Such circuits integrate a digital part, which involves some form of encryption, and an analog (RF) part, which transmits the encrypted data through a public wireless channel.

Compromising wireless cryptographic ICs

Within this context, we consider hardware Trojans whose payload objective is to leak secret information (i.e., the encryption key) over the wireless channel, so that the attacker can decipher the encrypted transmission. To maximize the utility of these hardware Trojans, given that the attacker may listen only to the public wireless channel and has no control over the circuit inputs, we assume that the Trojans are always active and do not rely on a particular trigger. In terms of type, these hardware Trojans are distributed across the circuitry that holds the secret information and the circuitry that controls the wireless transmission, and they manipulate only the parametric space but do not violate any of the design's functional specifications. Similar hardware Trojans aiming to leak the secret key through side channels have also been designed for digital cryptographic circuits.[7,8]

To our knowledge, ours is the first study of hardware Trojans in the mixed-signal SoC domain. Using as our experimentation vehicle a wireless cryptographic circuit and two example hardware Trojans that we designed for the purpose of this study, we demonstrate three key findings concerning attack complexity, detection difficulty, and a possible solution.

Attack complexity

Minor modifications to the digital part of a wireless cryptographic chip suffice to leak secret information without altering the far more sensitive analog part. The vulnerability of such chips stems partly from the fact that they transmit over a public wireless channel. Their true Achilles heel, however, is the fundamentally analog nature of wireless transmission, which entails several continuous parameters (e.g., amplitude, frequency, and phase). To tolerate variations due to fabrication process and/or operating conditions, specifications for these parameters are defined as windows of acceptable performances rather than exact values. As a result, a hardware Trojan can hide additional information within the tolerance margins of such continuous entities and secretly transmit it. Although such transmissions abide by all specifications and appear to be perfectly legitimate, an adversary who knows the structure of the additional information will be able to extract it.

Detection difficulty

Evading detection via traditional manufacturing test methods is trivial. The functionality of the digital part of the chip in normal operation mode and in test mode can be preserved despite the addition of the hardware Trojan; hence, no structural (i.e., scan-based) or functional tests, or even enhanced functional tests for hardware Trojan detection, will fail in a fault-free but Trojan-infested chip. Similarly, because the analog part of the chip is left intact, all analog and RF specification tests will pass. Furthermore, because the leaked information is hidden within the allowed transmission specification margins, system-level functional tests will also pass. We find, moreover, that existing side-channel fingerprint generation and checking methods, at least in their original form, fall short in detecting hardware Trojans in wireless cryptographic ICs.

Possible solution

Despite the fact that hardware Trojans can be hidden within the process variation margins of a wireless cryptographic chip and might not be exposed through any of the methods discussed so far, it might still be possible to detect them. Effective hardware Trojans must impose a specific structure on the transmission parameters, which the attacker leverages to snoop the secret key. Although the defender is unaware of this structure, advanced statistical analysis of these parameters might be sufficient to reveal its existence and, thereby, expose the hardware Trojan. Because the attacker does not know what data will be collected or how it will be analyzed, this method is difficult to evade. In other words, the element of surprise by the attacker, who picks the structure of the hidden data, is counteracted by a similar element of surprise by the defender, who picks the measurements and the statistical-analysis process.

Experimentation vehicle

Figure 1 shows the experimentation vehicle that we will use to elucidate these points. This is a

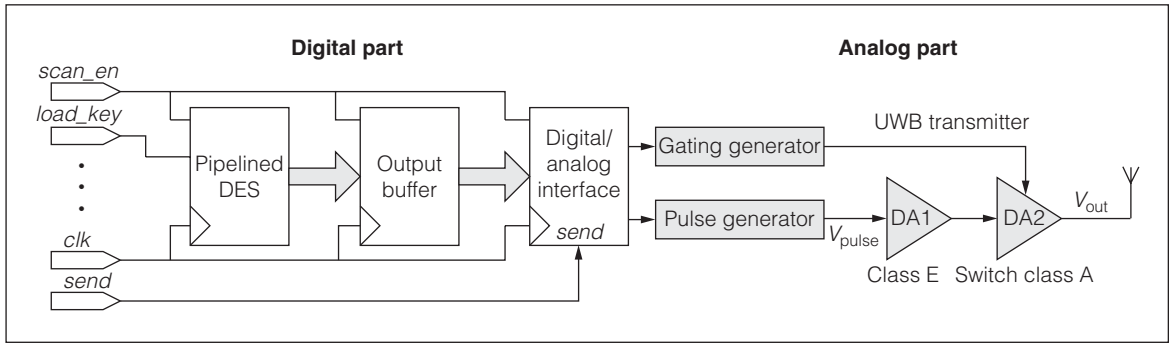

Figure 1. Block diagram of example wireless cryptographic IC.

mixed-signal wireless cryptographic IC, capable of encrypting and broadcasting data, which can be used in secure data transmission over open channels. The digital part includes a pipelined Digital Encryption Standard (DES) core (http://www.opencores.org/projects.cgi/web/des/overview), an output buffer, and a serializer that serves as the interface between the digital and the analog part. The analog part is an ultrawide-band (UWB) transmitter.

Implementation details

DES is widely used as a secret-key cipher algorithm, and its derivative, triple-DES, is quite popular in commercial security applications. The DES core in the chip is a performance-optimized design with 16 encryption blocks in a pipeline structure. Each block can independently run the Feistel function *f*, which is the central part of the DES algorithm. A fully pipelined key generation module is designed to operate in parallel with these encryption blocks. To achieve high operating frequency, the initial permutation and inverse initial permutation of the plaintext are handled through hard-wiring, with no logic circuitry involved. The widths of the input and output data are both 64 bits, which is the length of a plaintext or ciphertext block. The output buffer is a FIFO structure of 64-bit words, which supports reading and writing speeds commensurate with the performance of the pipelined DES core. The digital-to-analog interface converts the 64-bit data block from the buffer into a serial bit stream and passes it on to the UWB transmitter. The interface also adjusts the data-sending frequency to ensure signal integrity in this mixed-signal design. A pulse on the *send* primary input passes the contents of the output buffer to the interface and finally to the UWB transmitter for broadcasting.

UWB technology has gained wide attention since the US Federal Communications Commission (FCC) released an unlicensed spectrum of 3.1–10.6 GHz for commercial wireless applications. It can transmit data over a wide spectrum of frequency bands with very low power consumption and a high data rate. Our circuit integrates a UWB transmitter,[9] which consists of a pulse signal generator, a gating signal generator, and two driver amplifiers (DAs). The UWB transmitter is in active mode and transmits a high-frequency signal when the information bit to be transmitted is 1; otherwise, it is in idle mode. The full-CMOS design of this UWB transmitter makes it compatible with the digital part and helps further reduce overall power consumption and ultimately cost.

We designed the chip in TSMC CL013G 0.13-micron CMOS technology process (http://www.mosis.com/products/fab/vendors/tsmc/tsmc013-cl). The digital part runs at 75 MHz, and the UWB transmitter has a data rate that exceeds 50 Mbps. As commonly practiced in mixed-signal SoCs, test plans involve separate procedures for the digital and analog parts. For the digital part, these tests cover both stuck-at and delay faults using a full-scan chain of enhanced scan flip-flops. For the analog part, besides the traditional specification tests, we also measure the spectrum of the output pulse sequence of the DA chain at a data transmission rate of 50 Mbps.[9] In addition, functional tests at the system level involve a large amount of patterns that are randomly generated, encrypted, and broadcasted by the UWB transmitter. A receiver decrypts the ciphertext and compares it to the expected plaintext, to identify any discrepancies due to manufacturing faults.

Transmission specifications

Figure 2 shows a simulation example of a typical transmission of a 64-bit block of ciphertext and a magnified view of the transmission signal when a logical 1 is broadcasted. UWB specification calls for a

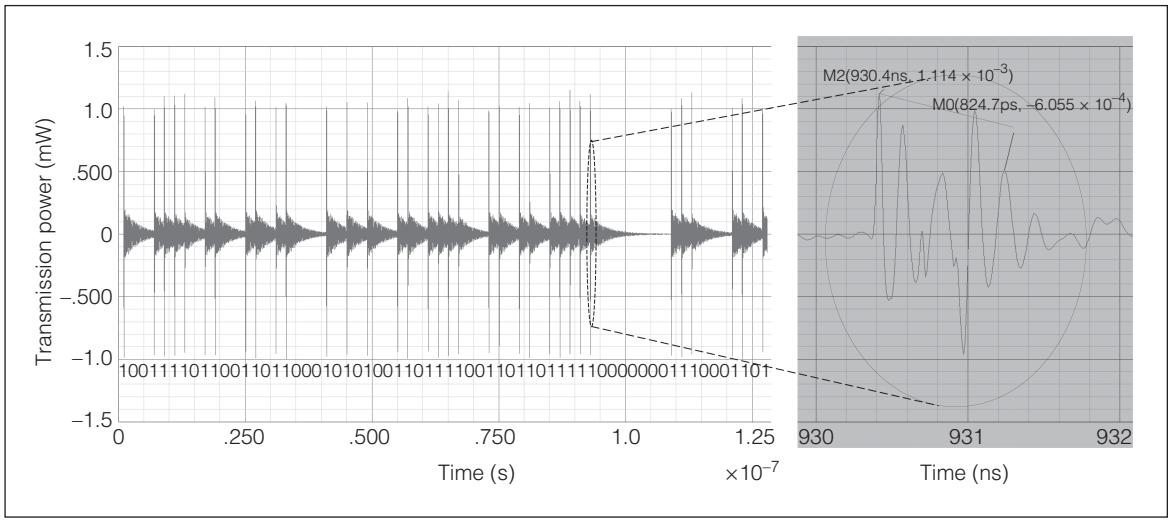

Figure 2. Example of a 64-bit ciphertext block transmission.

transmission frequency between 3.1 GHz and 10.6 GHz. The specifications for this particular UWB transmitter implementation define its frequency as being between 4 GHz and 6 GHz. Transmitting a logical 1 involves 5 to 7 peaks of amplitude over 300 μW with at least one of them over 900 μW. The actual performances of each individual chip will vary, depending on the fabrication process variations. For example, the response of the circuit instance shown in the figure, which was randomly picked from a population of 200 chips generated through a Monte Carlo Spice-level simulation with 5% process variation on all transistor parameters, operates at a frequency of 4.8 GHz and involves 5 peaks of amplitude greater than 300 μW, with the largest measuring at 1,114 μW.

Hardware Trojans

Here we describe two alternative hardware Trojans that can be incorporated in the example circuit. Through simple modifications on only the digital portion of the chip, these hardware Trojans leak the encryption key by hiding it in the wireless transmission amplitude or frequency margins allowed because of process variations; thus, they ensure that the circuit continues to comply with all of its functional specifications.

The working principle of these Trojans is simple: extract one bit at a time from the 56-bit encryption key, which is stored in the DES core, and leak it by hiding it in one 64-bit block of transmitted data. After only 56 ciphertext blocks are transmitted, the entire key will have been broadcasted, compromising the encryption.

Implementation details

Each hardware Trojan involves two modifications. The first modification (see Figure 3a) is common to both Trojans and aims to extract the encryption key from the DES core. The second modification (see Figure 3b) is different for each Trojan and aims to manipulate the transmission amplitude or frequency to leak the key through the wireless channel.

The key extraction modification exploits the ability of enhanced scan flip-flops to store two bits, one in the D flip-flop and one in the follow-up latch, so that back-to-back vectors can be applied for detecting delay faults when the circuit is in test mode. During normal operation, however, the latches are transparent, essentially holding the same information as the D flip-flops. In the example circuit, the 56-bit encryption key is stored in a sequence of 56 enhanced scan flip-flops that are serially connected in a scan chain, as shown in the top part of Figure 3a. The basic idea for extracting the secret key is to store it only in the latches of the enhanced scan flip-flops and reuse the D flip-flops to create a 56-bit rotator. Initially, when the user loads the key, both the flip-flops and the latches hold the correct bits. Then, every time a data block is transmitted, the last bit of this rotator is extracted and hidden in the transmission, while the rotator shifts its contents by one position. We emphasize that only the D flip-flops of the enhanced scan flip-flops hold a rotated version of the key, while the follow-up latches continue to hold the correct version, so that the ciphertext is correctly produced. Simple control logic consisting of a few gates, shown at the bottom of Figure 3a, suffices for this purpose.

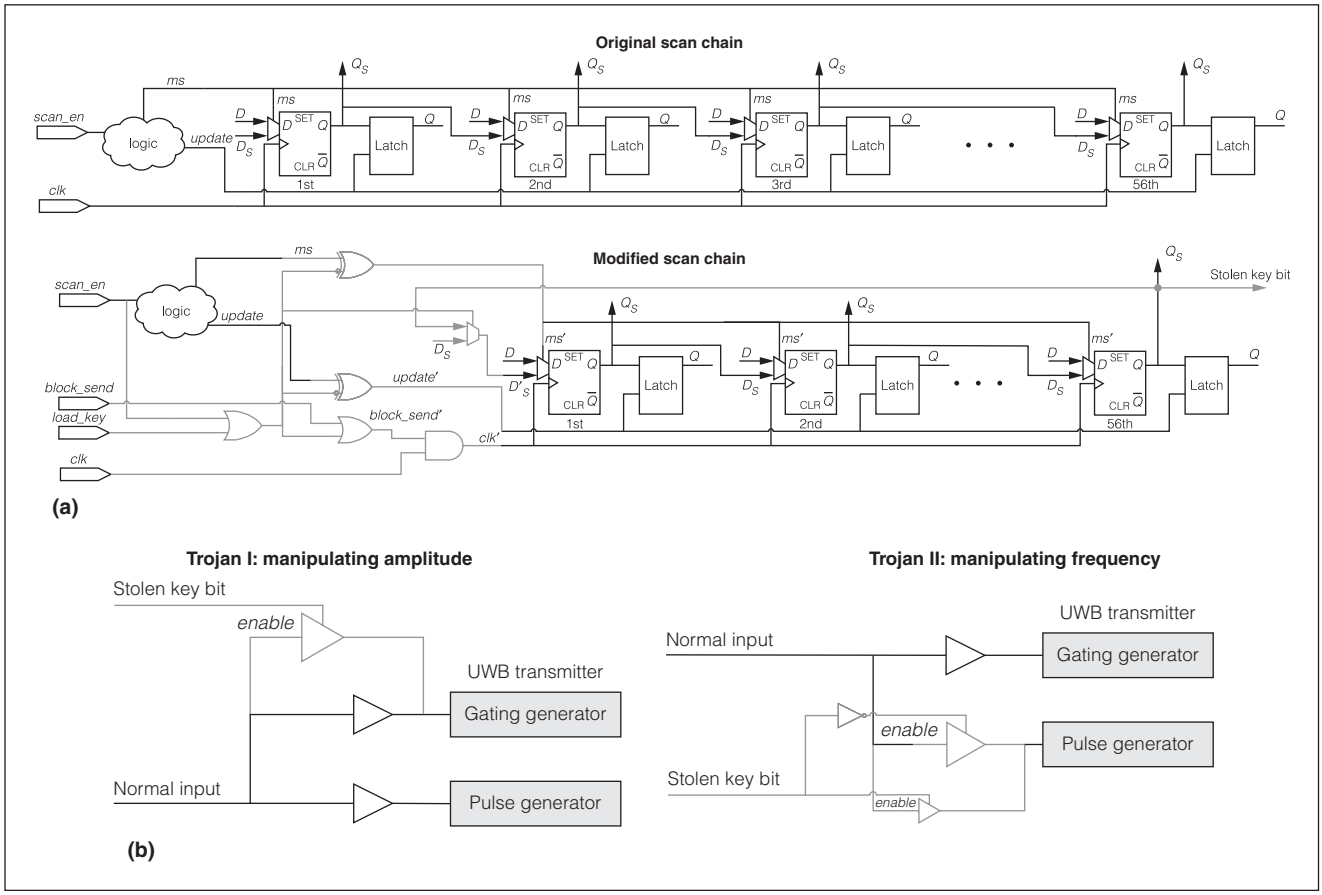

Figure 3. Extracting the key bitwise, through a rotator made of the 56 enhanced scan flip-flops where it is stored (a), and broadcasting the stolen key bit by manipulating the amplitude or the frequency of the UWB transmission (b). (Hardware modifications to Trojans shown in gray. ms: mode signal.)

The key transmission modification receives the stolen bit and modifies the transmission signal in one of two ways. The first option (Type-I), shown on the left side of Figure 3b, manipulates the transmission amplitude; when the stolen key bit is 1, an additional driver strengthens the legitimate transmission signal before it reaches the gating generator, thereby slightly increasing the transmission amplitude. Figure 4a shows the corresponding impact on the signal transmitted by the example circuit instance used in Figure 2. In this case, the amplitude increases from 1,114 μW to 1,235 μW, but the frequency remains at 4.8 GHz. The second option (Type-II), shown on the right side of Figure 3b, manipulates the transmission frequency; when the stolen key bit is 1, the original buffer is bypassed, and an alternative buffer is used to delay the output of the pulse generator, thereby slightly increasing the transmission frequency. Figure 4b shows the corresponding impact on the signal transmitted by the example circuit instance used in Figure 2. In this case, the frequency increases from 4.8 GHz to 5.2 GHz, but the amplitude remains at 1,105 μW. In both cases, when the stolen key bit is 0, no change occurs in the transmitted signal.

The overall area overhead incurred by each of these Trojans is around 0.02% of the digital part of the chip. Figure 4 assumes that the storage elements holding the secret key are enhanced scan flip-flops that are connected in sequence and already exist in the design for structural manufacturing test. If this is not the case, a separate 56-bit rotator made out of simple latches must be added to store and rotate the key so that it can be transmitted in a bitwise fashion. Even in that case, the area overhead is well below 0.4%, and the power overhead incurred in the worst case (when the key is a sequence of alternating 0s and 1s) is 0.25%.

Secret information extraction

Figures 4a and b show the transmission power waveform of a Type-I and a Type-II Trojan-infested chip, respectively, when the stolen key bit transmitted along with the legitimate signal is 1 and when it is 0. In the Type-I Trojan-infested chip, the difference in the stolen key bit value is reflected as a difference of 120 μW in the maximum amplitude. Similarly, in the Type-II Trojan-infested chip, the difference in the stolen key bit value is reflected as a 0.4-GHz difference in the frequency. Both of these differences are well within the margins allowed for process variations and operating-condition fluctuations and would not raise any suspicion. Although the attacker does not know a priori the exact amplitude or frequency levels in each of the two cases, the fact that this difference is always present suffices for extracting the secret key. All the attacker needs to do is listen to the wireless channel to observe these two different amplitude or frequency levels, which correspond to a stolen key bit of 1 and a stolen key bit of 0, respectively. Once these two levels are known, listening to 56 consecutive transmission blocks reveals a rotated version of the 56 bits of the encryption key. Using this information, the attacker needs at most 56 attempts (i.e., all possible rotations of the extracted 56 bits) to decrypt the transmitted ciphertext.

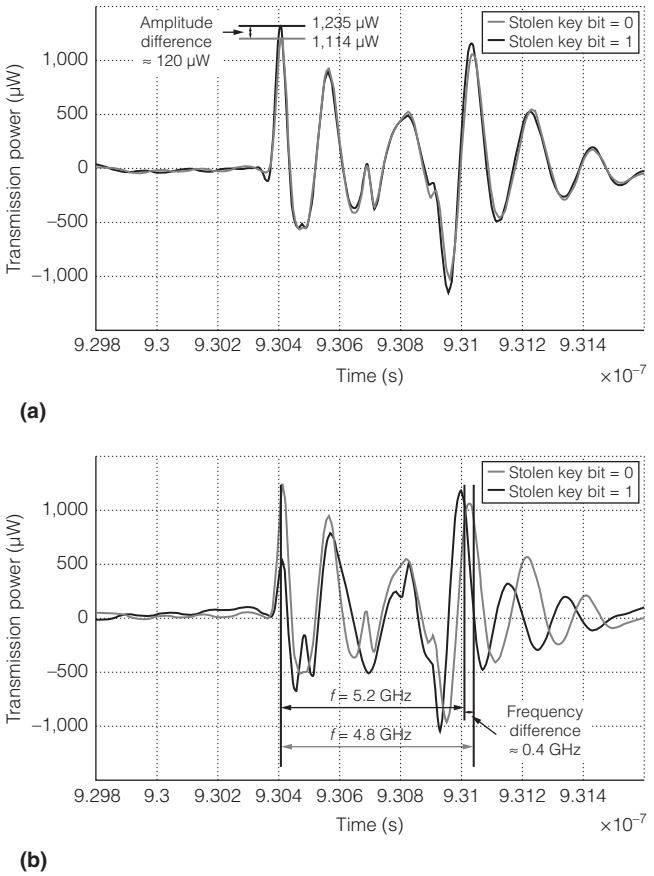

Figure 4. Difference in Type-I Trojan-infested circuit transmission depending on the value of the stolen key bit (a), and difference in Type-II Trojan-infested circuit transmission depending on the value of the stolen key bit (b).

Existing Trojan detection methods

The mechanism through which the two hardware Trojan examples leak the secret information over the wireless channel allows them to evade detection, not only by traditional manufacturing testing but also from previously proposed Trojan detection methods.

Functional, structural, and enhanced testing

Our hardware Trojan examples do not alter the functionality of the digital part of the circuit. In normal operation, the enhanced scan flip-flops that hold the key bits are loaded appropriately. We simulated numerous randomly generated functional-test vectors and verified the correctness of the produced ciphertext. In test mode, the scan chain also operates as expected. To demonstrate that structural tests do not detect these hardware Trojans, we used a standard industrial ATPG tool to generate test vectors for all stuck-at and delay faults in the Trojan-free circuit, and we simulated these tests on the two Trojan-infested circuits. As expected, all tests passed. Enhancing the test set with further vectors that exercise rare events is also ineffective,[3] because the hardware Trojans do not affect the digital functionality. The analog portion of the circuit is not modified and, therefore, also passes the traditional specification-based analog (RF) test.

System-level testing

System-level tests examining the parameters of the wireless transmission also fail to expose the hardware Trojans, because the structure added by the leaked information is hidden within the margins allowed for process variations. To demonstrate this, we measured the transmission power of 200 genuine (i.e., Trojan-free) chips, as well as 100 chips infested with a Type-I hardware Trojan and 100 chips infested with a Type-II hardware Trojan. All of these chips were generated using Monte Carlo Spice-level simulation assuming 5% process variations on all circuit

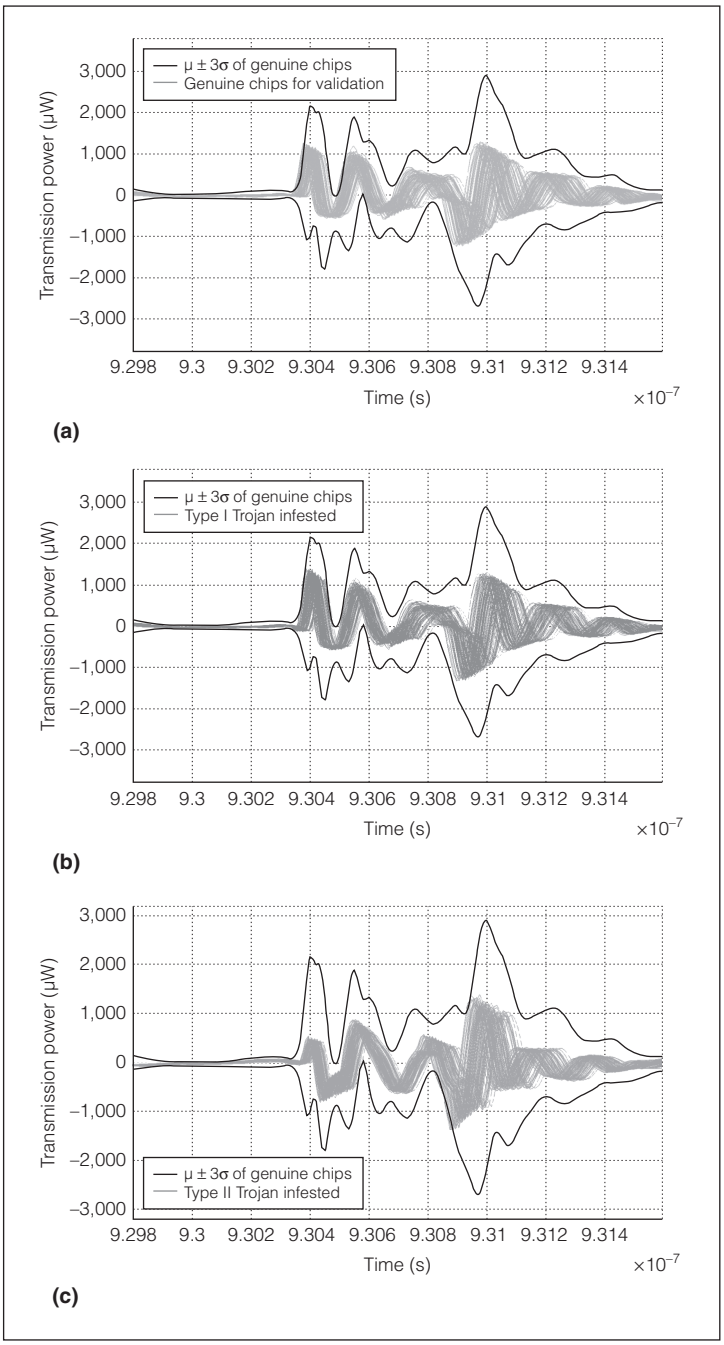

Figure 5. The $\mu \pm 3\sigma$ transmission power envelope of 100 Trojan-free chips and the transmission power of another 100 Trojan-free chips (a), the transmission power of 100 Type-I Trojan-infested chips (b), and the transmission power of 100 Type-II Trojan-infested chips (c).

parameters. Figure 5a plots the transmission power when a 1 is transmitted by half of the Trojan-free chips, as well as the $\mu \pm 3\sigma$ envelope of the transmission power when a 1 is transmitted by the other half of the Trojan-free chips. Figures 5b and c plot the transmission power when a 1 is transmitted by the Type-I and Type-II Trojan-infested chips, respectively. Clearly, given any one of these transmission power plots, it is not possible to distinguish whether it comes from a Trojan-free or a Trojan-infested chip.

Local current traces

An interesting hardware Trojan detection method was recently developed on the basis of local current traces.[2,6] This test strategy detects anomalies introduced by the Trojan in the currents measured at the power ports and takes into account process and operating-condition variations. The authors demonstrated that their method can detect Trojans as small as 2% of the power grid. To implement this method in our design, we would have to divide the chip into at least 20 power grids with at least 30 uniformly located power ports. The availability of these power ports is a serious obstacle to implementing this method. Furthermore, a capable attacker would probably observe the existence of these power ports and could possibly invent countermeasures to prevent the injected hardware Trojans from becoming visible through these ports.

Global power traces

Another method, global power consumption traces, has been applied to distinguish between Trojan-free and Trojan-infested chips.[4] The method employs statistical analysis of the eigenvalue spectrum and can effectively detect hardware Trojans occupying 0.12% of the total circuit area, assuming process variation in the order of 5%. But when the hardware Trojan area is reduced to only 0.01% and the process variation is increased to 7.5%, false alarms start to appear. Considering the very low area overhead of our hardware Trojans (0.02%), and based on the limitations of the traces,[4] we do not expect that statistical analysis of the total power consumption will expose them. Indeed, even when we applied this method only to the power traces of the digital part (mixed-signal SoCs typically have separate power ports for the analog and digital parts), wherein the hardware Trojans are hidden, it was not possible to effectively distinguish between Trojan-free and Trojan-infested chips in any eigenvalue subspace. Nevertheless, other parameters might still prove effective.[4] In fact, the solution that we propose employs a similar statistical analysis of the wireless transmission power.

Path delay traces

A similar statistical method uses path delay fingerprints to differentiate Trojan-free from Trojan-infested chips.[5] Although our hardware Trojan examples add some delay to a few paths in the digital part of the circuit, the impact is too small to be observed. Even if we knew which paths to check (i.e., those related to the encryption key), the complexity of the pipelined encryption circuitry would provide enough margin to hide the added delay. To verify this, we applied the path delay traces method assuming process variations in the range of 5%, only to discover its inability to distinguish the Trojan-free from the Trojan-infested chip populations.

Statistical analysis to the rescue

For the attacker to extract the stolen key, hardware Trojans must add some form of structure to the transmitted signal. In our examples, this structure corresponds to an increase in the amplitude or the frequency of transmission when the stolen key bit has a value of 1. Although the individual transmissions are within the acceptable specification boundaries, this added structure enables the possibility that such hardware Trojans can be exposed through statistical analysis of the transmission parameters.

To demonstrate this principle, we use as a measurement the total transmission power for broadcasting one block of data (64 bits). For the 100 Type-I Trojan-infested, 100 Type-II Trojan-infested, and half of the 200 Trojan-free circuit instances that we generated via Monte Carlo Spice-level simulation with 5% process variations, as we already discussed, we measure the total transmission power when transmitting each of six randomly selected blocks (the same for all circuits). Of course, the Trojan-infested chips also leak one key bit during each of the six transmissions, half of which are set to 1. All six measurements for all genuine and all Trojan-infested chip populations are within the acceptable specification range. Even when we project the three chip populations on the 6D space of these measurements, we cannot distinguish them because they fall upon one another. Although we cannot pictorially illustrate this in six dimensions, Figure 6a shows a projection of the three populations on three of these dimensions. Clearly, separating the genuine from the Trojan-infested populations in this space is not possible. The situation is similar for any other subset of three measurements.

However, running a principal component analysis (PCA) on these measurements reveals that the structure of the genuine-chip data is different from the structure of the Trojan-infested-chip data.[10] Figure 6b shows a projection of the three populations on the data's three principal components, clearly revealing that they are separable in this space. Therefore, we can define the trusted boundary as a simple minimum volume ellipsoid (http://www.seas.upenn.edu/~nima/papers/Mim_vol_ellipse.pdf) that encloses the genuine population. Then, we can discard as suspicious any chip whose footprint on the space of the selected three principal components does not fall within the trusted boundary. In our example, this method detects all Type-I and Type-II Trojan-infested chips without inadvertently discarding any genuine chips. Indeed, to verify this last point, we projected the footprint of the remaining 100 Trojan-free circuit instances on the space of the selected three principal components, and none of them fell outside the shown ellipsoid.

We conclude by pinpointing the fundamental reason why such statistical analysis is effective. The attacker's arsenal includes the ability to pick the structure of the leaked information and the ability to hide the effect of the hardware Trojan within the allowed margins. On the other hand, the defender's arsenal includes the ability to pick the actual measurements and the statistical analysis process. Given the small number of transmission parameters (or combinations thereof) wherein the attacker can hide the added structure, as well as the large number of measurements that the defender can use to identify statistical discrepancies, we believe that the odds are in favor of the defender. Finally, we should mention that similar statistical analysis and machine-learning-based methods have been employed successfully for the purpose of manufacturing testing[11] and radiometric fingerprinting[12] of analog and RF circuits. Yet, to our knowledge, this is the first attempt to apply such methods to hardware Trojan detection in wireless cryptographic ICs.

HARDWARE TROJAN ATTACKS on wireless cryptographic ICs constitute a tangible threat, capable of compromising the applications where such chips are deployed. Although such hardware Trojans openly broadcast secret information, such as an encryption key, through an additional structure that is carefully hidden within the legitimate transmission,

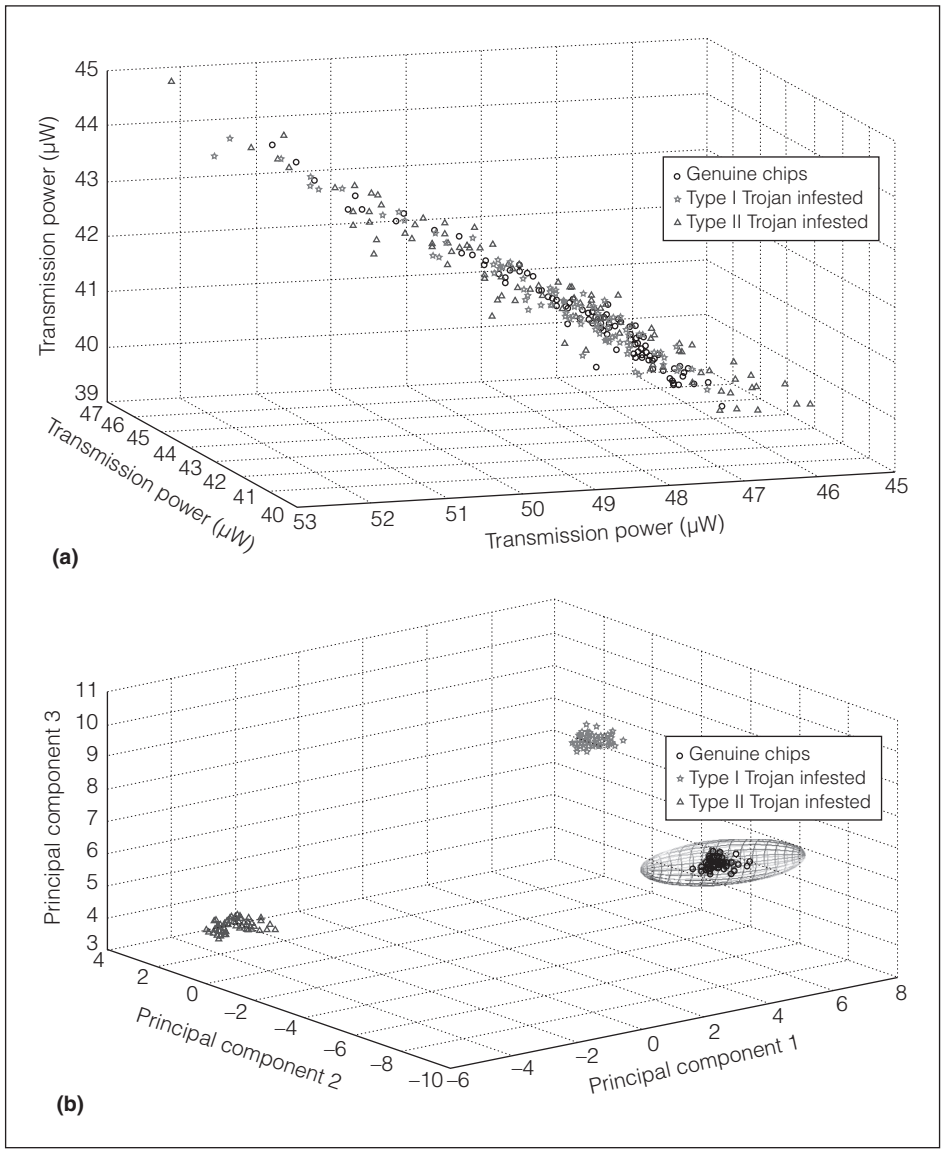

Figure 6. Projection of genuine and Trojan-infested chip populations on three out of six transmission power measurements (a), and projection of genuine and Trojan-infested chip populations on three principal components of six transmission power measurements (b).

traditional manufacturing testing and existing hardware Trojan detection methods fail to expose them. Even though this added structure is known only to the attacker, its sheer existence provides the basis for a hardware Trojan detection method that is particularly geared toward wireless cryptographic ICs. Our conjecture, based on the experimental evidence obtained through the example circuit and hardware Trojan designs, is that a statistical analysis of the various transmission parameters of a wireless cryptographic IC can effectively detect chips that leak additional information.

Our future plans involve fabrication of Trojan-free and Trojan-infested versions of the example circuit and validation of our findings using actual chip measurements as opposed to simulations. Although the study we have discussed here considers only one example circuit and two types of hardware Trojans, it paves the way for further experimentation with more complicated wireless cryptographic ICs and more sophisticated Trojan designs. ■

■ References

1. S. Adee, "The Hunt for the Kill Switch," *IEEE Spectrum,* vol. 45, no. 5, 2008, pp. 34-39.
2. R.M. Rad et al., "Power Supply Signal Calibration Techniques for Improving Detection Resolution to Hardware Trojans," *Proc. IEEE/ACM Int'l Conf. Computer-Aided Design* (ICCAD 08), IEEE CS Press, 2008, pp. 632-639.
3. F. Wolff et al., "Towards Trojan-Free Trusted ICs: Problem Analysis and Detection Scheme," *Proc. IEEE Design Automation and Test in Europe* (DATE 08), IEEE CS Press, 2008, pp. 1362-1365.
4. D. Agrawal et al., "Trojan Detection Using IC Fingerprinting," *Proc. IEEE Symp. Security and Privacy,* IEEE CS Press, 2007, pp. 296-310.
5. Y. Jin and Y. Makris, "Hardware Trojan Detection Using Path Delay Fingerprint," *Proc. IEEE Int'l Workshop Hardware-Oriented Security and Trust,* IEEE CS Press, 2008, pp. 51-57.
6. R. Rad, J. Plusquellic, and M. Tehranipoor, "Sensitivity Analysis to Hardware Trojans Using Power Supply Transient Signals," *Proc. IEEE Int'l Workshop Hardware-Oriented Security and Trust,* IEEE CS Press, 2008, pp. 3-7.
7. Y. Jin, N. Kupp, and Y. Makris, "Experiences in Hardware Trojan Design and Implementation," *Proc. IEEE Int'l Workshop Hardware-Oriented Security and Trust,* IEEE CS Press, 2009, pp. 50-57.

8. L. Lin et al., "Trojan Side-Channels: Lightweight Hardware Trojans through Side-Channel Engineering," *Proc. Cryptographic Hardware and Embedded Systems,* LNCS 5747, Springer-Verlag, 2009, pp. 382-395.
9. T. Yuan et al., "A Fully Integrated CMOS Transmitter for Ultra-wideband Applications," *Proc. IEEE Radio Frequency Integrated Circuits Symp.,* IEEE Press, 2007, pp. 39-42.
10. I.T. Joliffe, *Principal Component Analysis,* Springer-Verlag, 1986.
11. H.G. Stratigopoulos and Y. Makris, "Error Moderation in Low-Cost Machine-Learning-Based Analog/RF Testing," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,* vol. 27, no. 2, 2008, pp. 339-351.
12. A. Candore, O. Kocabas, and F. Koushanfar, "Robust Stable Radiometric Fingerprinting for Frequency Reconfigurable Devices," *Proc. IEEE Int'l Workshop Hardware-Oriented Security and Trust,* IEEE CS Press, 2009, pp. 43-49.

Yier Jin is a doctoral candidate in electrical engineering at Yale University. His research interests include reliable, secure, and trustworthy ICs, as well as dynamically reconfigurable architectures. He has an MS in electrical engineering from Zhejiang University, Hangzhou, China. He is a student member of the IEEE.

Yiorgos Makris is an associate professor of electrical engineering at Yale University. His research focuses on applications of machine learning in test, reliability, and security of ICs. He has a PhD in computer engineering from the University of California, San Diego. He is a senior member of the IEEE.

■ Direct questions and comments about this article to Yier Jin and to Yiorgos Makris, Dept. of Electrical Engineering, Yale University, PO Box 208267, New Haven, CT 06520-8267; yier.jin@yale.edu and yiorgos.makris@yale.edu.