Runtime Detection of Probing/Tampering on Interconnecting Buses

Zhenyu Xu, Thomas Mauldin, Qing Yang and Tao Wei
Electrical, Computer, and Biomedical Engineering, University of Rhode Island
Kingston, RI 02881, USA
Email: {zhenyu_xu, thomas_mauldin, qyang, tao_wei}@uri.edu

***Abstract*—It has been reported that physical probing on an off-chip bus can reveal confidential information in an electronic system. An attacker can use non-invasive and inexpensive electric probes (or interposers) to measure signals from circuit traces, such as the memory bus between the memory controller and a memory module. This paper describes a method to detect any bus probing/tampering by tracking the phase shift of output digital waveforms, induced by input impedance change at the bus transmitter (Tx). A low-overhead digital circuit based on flip-flop's metastability is built around the Tx using a field-programmable logic gate array (FPGA) to precisely measure the phase shift of output signals. Uniquely, the output data launched by the Tx is used as a stimulus signal, thus, the proposed method holds the advantage of detecting probing attacks at runtime. That is, the detection action operates in parallel with the normal data transfer on a bus without any interference, imposing zero latency to the communication channel. In order to show its feasibility in a real-world communication protocol, we implemented the proposed method in the DDR memory controller on an FPGA board (Xilinx ZCU104). The working prototype is able to protect a memory bus between the FPGA board and a DDR4 DIMM with a data rate of 2400MT/s. Experimental results show that the proposed method can be used to countermeasure interposer attacks, probing attacks, and cold boot attacks. We believe that the proposed method can be implemented in a variety of communication channels.**

I. Introduction

Physical probing/tampering on an interconnecting bus can reveal confidential information of data being transferred from one chip to another. An adversary can use low-cost probes or specially designed probes (interposers) together with a logic analyzer or an oscilloscope to eavesdrop data busses. For example, it has been demonstrated that one can launch a Direct Memory Access (DMA) attack to a Dual In-line Memory Module (DIMM) via an interposer and logic analyzer [1]. The classic solution to protect data privacy during transfer is data encryption. For example, modern in-vehicle Controller Area Network (CAN) bus communication uses encryption to protect against sniffing attacks [2]. Another example is Intel Software Guard Extension (SGX) Memory Encryption Engine (MEE), which provides cryptographic protection of external memories [3]. MEE is a hardware encryption/decryption engine, which operates under the assumption that the security perimeter includes only the internals of the CPU package and leaves the DRAM untrusted (subject to physical probing). MEE protects the confidentiality of the CPU-DRAM traffic. A similar concept has been adopted to protect field-programmable logic gate array (FPGA) to DRAM channels [4]. However, the grand challenge of encryption on high-speed busses is very high overhead in terms of extra latency and power consumption. This is especially painful for external Double Data Rate (DDR) memory buses, where full encryption requires every write/read transaction to be encrypted/decrypted at a high clock rate. Thus, most encryption solutions, such as Intel MEE and AMD Secure Memory Encryption (SME), performs partial DDR memory protection [5]. In addition, memory encryption only encrypts data, but not address, since DDR chips do not have a decryption engine. Also, even if DDR chips are equipped with a decryption engine, securely passing the encryption key from memory controller to DDR chips is challenging. A recently reported attack, namely Membuster, exploits the address bus on a memory bus, and successfully defeated Intel SGX MEE [6].

Very recently, several methods, aiming at protecting buses at its physical layer, were proposed. Oksman proposes to detect changes in the impedance of the DRAM bus caused by probing/tampering [6]. The impedance changes are indirectly measured by introducing controlled DRAM write errors. The data strobe (DQS) signal and its associated data (DQ) signal are purposefully misaligned, which creates data writes that store different values to the memory. By statistically evaluating the data errors stored in the DDR, this method is able to detect modifications of the DRAM bus, such as an added interposer between the memory chip and an FPGA board. However, this method requires to halt the entire system to perform detection, adding significant latency and making it impossible for runtime protection. Xu et al. proposes to build an integrated time-domain reflectometer in a bus I/O interface to extract an impedance profile over distance, which is a direct measure of bus probing/tampering and its location along the trace [7]. Uniquely, this method uses the data as a stimulus signal, making runtime detection possible. In addition, the extracted impedance pattern can be used as a Physical Unclonable Function (PUF) to authenticate a bus. However, this method is complicated to implement, and it requires the use of several external circuit components, limiting its broader application.

This paper describes a new method to detect bus probing/tampering by tracking the phase shift of output waveform at the transmitter (Tx) side, induced by input impedance

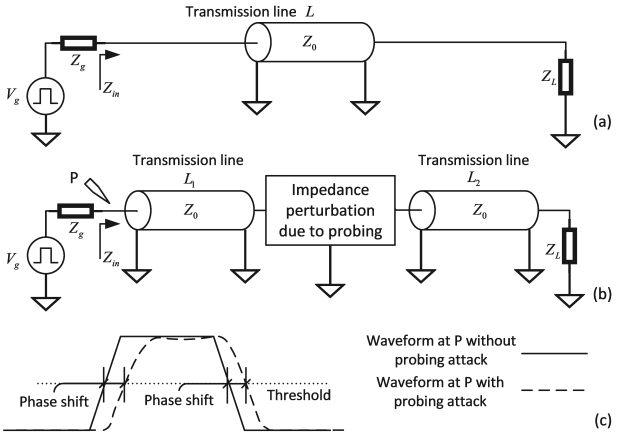

Fig. 1. Working principle on a single-ended bus. (a),(b): circuit models without and with probing attack; (c): waveforms without and with probing attack.

change of a bus. The input impedance change occurs essentially as a result of bus probing/tampering. We are able to catch such very small impedance change by exploiting the inherent metastability of flip-flops (FFs). Our design is surprisingly simple using existing digital circuits that can potentially track this phase shift [8], [9]. Our new design can be implemented using field-programmable logic around the bus Tx to precisely measure and track the phase shift of output signals. Uniquely, the output digital waveforms launched by the Tx are used as stimulus signals, and hence the proposed method holds the advantage of detecting probing attacks at runtime. That is, probe detection is done concurrently with the normal data transfer on a bus without any interference, and the detection action does not require stopping the normal data transfer, imposing zero latency to the communication channel.

In order to show its feasibility in a real-world communication protocol, we implemented the proposed method in a DDR memory controller on an FPGA board (Xilinx ZCU104). Our working prototype is able to protect the memory bus between the FPGA board and a DDR4 DIMM with a data rate of 2400MT/s. A series of bus probing/tampering attacks were launched on the memory bus, including regular probing attack, add-on interposer attack, and cold boot attack, to verify the effectiveness of the proposed method at runtime.

II. Working Principle

A. Phase Shift due to Impedance Change

A single-ended communication channel, i.e. a bus (clock, address, control, data, chip etc.), can be modeled as a simple circuit shown in Fig. 1(a). V_g is the input voltage, Z_g is the output impedance, or generator impedance, and Z_L is the load impedance at the receiver (Rx) end. A transmission line (Tx-line) with a length of L and a characteristic impedance of Z_0 is connecting the Tx, also referred to as bus driver or generator, to the Rx. An attack probe introduces an impedance perturbation to this bus at a point along the bus, also shown in Fig. 1(b), where $L = L_1 + L_2$. For example, an electric

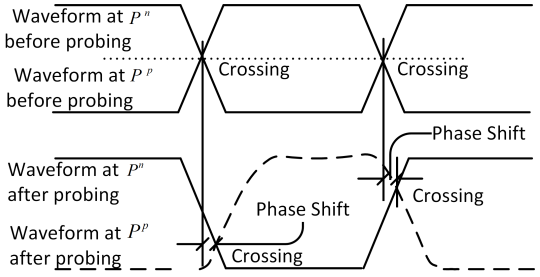

Fig. 2. Illustration of probing attack induced phase shift on a differential bus.

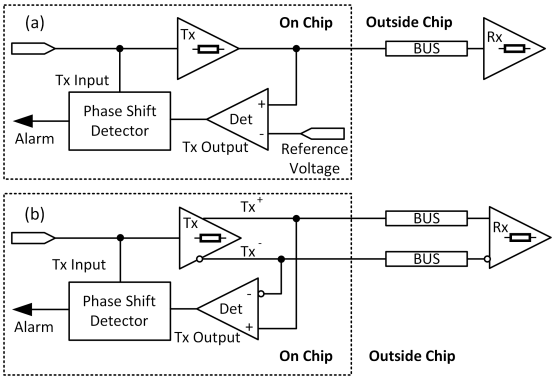

Fig. 3. Block diagram of the detection circuit; (a) single-ended bus; (b) differential bus.

probe touches the trace and introduces a shunt capacitance. It is worth noting that a contact probe also introduces serial resistance change, and an add-on bus interposer may introduce a more complex impedance perturbation. In addition, temperature gradient can also cause a impedance variation. Input impedance (Z_{in}) is the equivalent impedance right outside the Tx, looking into the Tx-line. Probing/tampering attempts unavoidably induce Z_{in} change, leading to a phase shift on output signals. Fig. 1(c) illustrates this concept. Waveforms at the Tx output (point P) without and with a probing attack are shown, where a phase shift is observed. Thus, the phase shift of the Tx output is a direct indicator of bus probing. This concept also applies to a differential bus. Fig. 2 illustrates the situation on a differential pair. A differential bus can be modeled with a pair of identical Tx-lines with identical load and generator impedance. P^n and P^p are physical points right outside the differential Tx on negative and positive sides, respectively. Assuming that a contact probe touches the positive trace, a phase shift is observed on the waveform at the positive output of the Tx (P^p), shown in Fig. 2. Effectively, the crossing-points experience a phase shift, which is a clear indicator of probing attack.

B. Detection Circuit

Fig. 3(a) shows the block diagram of the proposed probing detection circuit around a bus Tx. The original channel includes Tx, bus (a PCB trace), and Rx. The Tx is inside an IC chip, while the bus is on the PCB. The probe detection

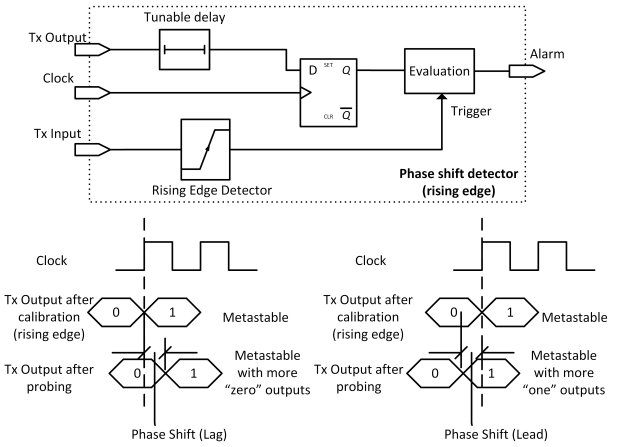

Fig. 4. Working principle of metastability-based phase shift detector (rising edge).

circuit is built around the Tx. It mainly consists two modules: a detector (Det, essentially the input buffer associated with the bidirectional I/O), and a phase shift detector (detailed in the subsection C). When this system is in use, the Tx receives digital signals, referred to as Tx input, from an internal circuit. The Tx drives the bus and launches the signal into the bus. Next to the Tx, the input buffer (serves as Det) converts the Tx output waveform into the internal digital format, referred to as Tx output. The reference voltage is set in between the high and low voltage levels of the Tx output. It is worth noting that this architecture is readily available in a bidirectional digital I/O interface. The phase shift detector tracks the phase shift of the Tx output at the rising or/and falling edges of data in transmission. Once an abrupt phase shift is detected, it sends out an alarm signal to the system. The detection circuit for a differential bus Tx can be built in a similar fashion, shown in Fig. 3(b). The difference is that the Det captures the crossing point of positive and negative waveforms at the output port of positive and negative Txs. The Det output is referred to as Tx output.

C. Metastability-based Phase Shift Detector

Potentially, any high-resolution delay measurement circuit can be used for phase shift detection. Here, we propose a simple and noise-insensitive metastability-based phase shift detector. Fig. 4 shows the design, which detects probe-induced phase shift change of the data's rising edge. The Tx output goes through a tunable delay line, and is captured by an FF, which is synchronized with the system clock. A rising edge detector is used to identify incoming rising edges (transition from "0" to "1"). At the system initialization stage, the tunable delay line is adjusted to align the rising edge transition with the rising edge of the clock. This way, the FF is working under metastable state, in which it could output either "0" or "1". In addition, the random noise on the circuit, stemmed from thermal noise and jitter, may randomly bias the output of the FF to output (denoted as Q) "0" or "1". The output of the FF is statistically stable, i.e., the probability of outputting "1", $P\{Q = 1\}$, is a constant. This probability, rate of logic "1" over a fixed number of tests (denoted as ρ), is measured using the evaluation module, which counts the number of captured "1"s (denoted as M) over a certain number of rising edges (denoted as N). Thus, $\rho = M/N$. During system initialization, the delay line is automatically tuned so that ρ is close to 50%. This delay (denoted as T_d) is saved in the system. During runtime operation, upon probing/tampering attack, the phase shift moves Tx output. For example, shown in the bottom left of Fig. 4, Tx output is shifted to the right (lag). As a result, the ρ decreases. In this case, the phase shift detector launches an alarm signal. The bottom right figure shows the "lead" case, in which the phase of Tx output is shifted to the left, and the ρ increases. Similarly, phase shift detector can function upon falling edges.

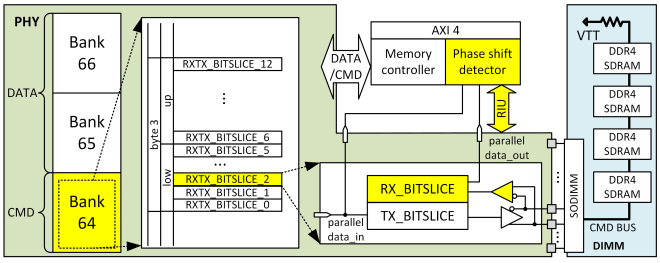

Fig. 5. Implementation on the DDR4 memory controller.

III. Experiment

A. FPGA Implementation

The proposed method was implemented in a DDR4 memory controller on a Xilinx FPGA development board (ZCU104). A DIMM (Micron MTA4ATF51264HZ-2G6E1) with four DDR4 chips and a total storage capacity of 2GB was mounted on the memory slot of the ZCU104. A Xilinx Memory Interface Generator (MIG) is employed to generate the DDR4 memory controller including the logic for physical interface (PHY). The DDR is configured to operate with a data rate of 2400MT/s and a clock speed of 1200MHz. We first verified that the system can write/read data into/from the DIMM seamlessly. Then, we implemented our design in the DDR memory controller by modifying the PHY, shown in Fig. 5 (modifications highlighted in yellow). Our goal is to build the detection circuit in one lane of the memory bus to demonstrate the proposed idea. Clock lane (differential pair), which belongs to command/address (CMD/ADDR) bus cluster (fly-by topology on DIMM), was protected. The clock I/O was modified from unidirectional to bidirectional in order to monitor the output waveform. PHY Register interface unit (RIU) was connected to Processing Unit (PS), allowing us to tune the input delay of any bitslice, as described in Fig. 4, using software. The phase shift detector was built on Programmable logic (PL) fabric. It triggers on the rising edge of signal (clock lane), records $\rho = M/N (N = 10^4$ in our design), and sends it to the PS. The PS was also employed to control the detection circuit and write/read data into/from the DDR. A memory test program was running

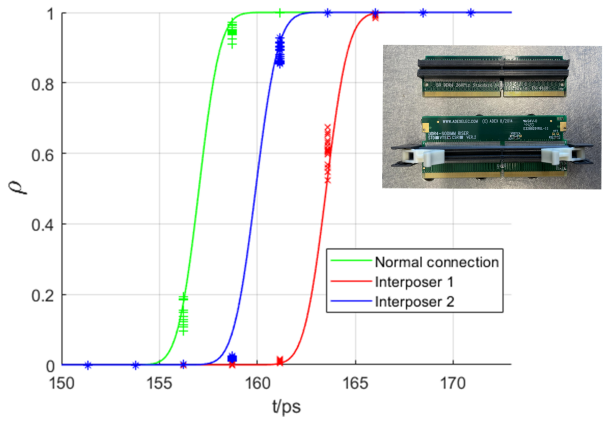

Fig. 6. Rate of logic "1", ρ, over different input delays.

during the entire experiments to make sure the memory controller was under normal operation, allowing us to verify and demonstrate the runtime probing/tampering detection using the proposed method.

B. Add-on Interposer Detection

It has been reported that an add-on DIMM interposer together with a logic analyzer or an FPGA can be employed to breach secured systems, such as Intel SGX MEE [1], [6]. We designed an experiment to show that our method is effective in protecting a bus against such attacks. Shown in Fig. 6, the capturing rate of "1", ρ, is plotted against different input time delay points, where delay adjustment resolution is 2.44ps. It clearly shows the FF's metastability transition. For example, under normal connection, the transition is between 156ps and 159ps. Two different interposers were added between the board and DIMM to test the detection circuit. The phase shift of Tx' output signal is a direct indicator of add-on interposers.

C. Probing Detection

An active probe $(100k\Omega, 0.6pF)$ was applied on the clock lane to emulate a probing attack as shown in Fig. 7. Probing location 1 is next to the first DDR chip and the location 2 is at the termination of the clock lane in CMD/ADDR cluster. Fig. 7(a) plots the ρ against input time delay points. It clearly shows that both probing attempts resulted in significant phase shift or Z_{in} change. By fixing the input time delay at 156.2ps, one can plot the histogram of ρ, indicating that the change of ρ is a direct indicator of probing attacks.

D. Cold Boot Detection

A cold boot attack relies on the data remanence property of DRAM to retrieve memory contents that remain readable in the minutes after power has been removed at low temperatures [10]. A freeze spay was used to quickly bring down the temperature of the DIMM to emulate cold boot attack. We observed that the sudden drop of the ambient temperature also induced an abrupt Z_{in} change, leading to a detectable phase shift. Fig. 8 plots the distribution of ρ (at a fixed input

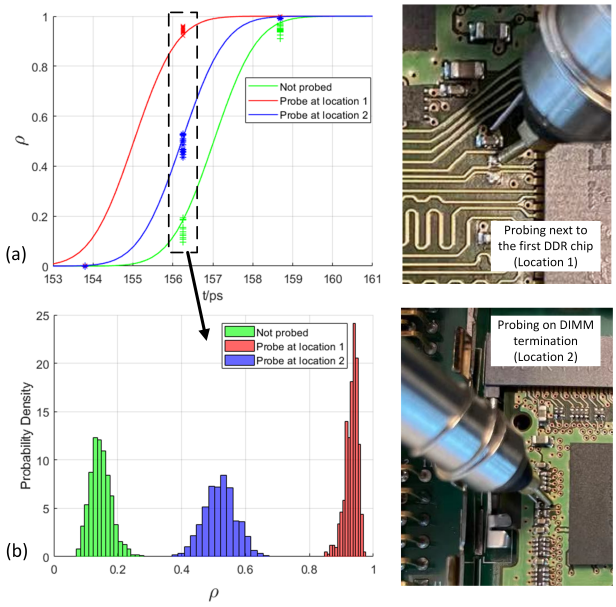

Fig. 7. Rate of logic "1", ρ, over different input delays; Probability density against ρ.

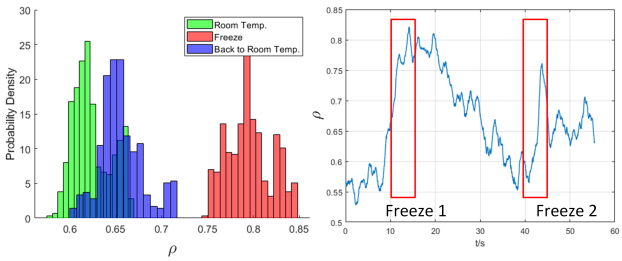

Fig. 8. Histogram of ρ under room temperature, during cold boot attack, and back to room temperature; ρ over time.

time delay point) under normal operation, cold boot attack, and back to normal, as well as ρ over time during cold boot attack. It clearly showed that the cold boot attack can be detected at runtime.

IV. Conclusion

This paper presents a method to catch bus probing/tampering by tracking the phase shift of output digital waveform, induced by Z_{in} change at the bus transmitter (Tx). A low-overhead and highly scalable digital circuit based on FF's metastability is proposed to precisely track the phase shift. The digital waveform launched by the Tx is used as a stimulus signal, allowing runtime detection of probing attacks. We implemented the proposed method in a DDR memory controller on an FPGA board at a data rate of 2400MT/s. Experimental results show that the proposed method can be used to countermeasure interposer attacks, probing attacks, and cold boot attacks. We believe this method can be used for other communication protocols, including Ethernet bus, Peripheral Component Interconnect Express (PCIe) bus, interposer/bus in chiplets, etc.

REFERENCES

[1] A. Trikalinou and D. Lake, "Taking DMA Attacks to the Next Level," *Black Hat USA*, 2017.

[2] P. Kleberger, T. Olovsson, and E. Jonsson, "Security aspects of the in-vehicle network in the connected car," in *IEEE Intelligent Vehicles Symposium, Proceedings*. IEEE, 2011, pp. 528–533.

[3] S. Gueron, "Memory encryption for general-purpose processors," *IEEE Security and Privacy*, vol. 14, no. 6, pp. 54–62, 2016.

[4] M. Werner, T. Unterluggauer, R. Schilling, D. Schaffenrath, and S. Mangard, "Transparent memory encryption and authentication," in *2017 27th International Conference on Field Programmable Logic and Applications, FPL 2017*. IEEE, 2017, pp. 1–6.

[5] D. Kaplan, J. Powell, and T. Woller, "AMD Memory Encryption," *White Paper*, 2016. [Online]. Available: http://amd-dev.wpengine.netdna-cdn.com/wordpress/media/2013/12/AMD_Memory_Encryption_Whitepaper_v7-Public.pdf

[6] A. Oksman, "A Method for Detecting DRAM Bus Tampering," Thesis, Aalto University, jan. 2020.

[7] Z. Xu, T. Mauldin, Z. Yao, S. Pei, T. Wei, and Q. Yang, "A Bus Authentication and Anti-Probing Architecture Extending Hardware Trusted Computing Base off CPU Chips and beyond," *Proceedings - International Symposium on Computer Architecture*, vol. 2020-May, pp. 749–761, 2020.

[8] M. A. Daigneault and J. P. David, "A high-resolution time-to-digital converter on FPGA using dynamic reconfiguration," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 6, pp. 2070–2079, 2011.

[9] C. Favi and E. Charbon, "A 17ps time-to-digital converter implemented in 65nm technology," in *Proceedings of the 7th ACM SIGDA International Symposium on Field-Programmable Gate Arrays, FPGA'09*, 2009, pp. 113–120.

[10] J. Alex Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten, "Lest we remember: Cold boot attacks on encryption keys," *Proceedings of the 17th USENIX Security Symposium*, vol. 52, no. 5, pp. 45–58, 2008.