Secure Intermittent Computing

Archanaa S. Krishnan, Charles Suslowicz, Daniel Dinu, and Patrick Schaumont

***Abstract*—Intermittent systems are a type of embedded device that is capable of preserving its state across periods of power loss and resuming execution from a previously saved state upon restoration of power. This capability is often accomplished through the use of system checkpoints that record the vital information about the current system state to restore operation once power is available. This demonstration illustrates the capability to protect such an intermittent system from an attacker that may attempt to scan, alter, or replay sensitive program state stored during periods of power loss. Observers will be able to inspect the current system state, checkpoint values, and computation progress while completely controlling the power supply to the target system. The audience will see it is not possible to create multiple checkpoints with identical state, reorder checkpoints, or replay the same execution multiple times. This functionality is a key requirement for the security of intermittent systems as their employment grows throughout the Internet-of-Things.**

I. Introduction

Intermittent systems are a growing area of embedded computing and present a unique set of challenges for proper security. Their capability to record the vital state information of the system for restoration after a period of power loss provides a ripe opportunity for adversaries to affect sensitive operations that span periods of power loss, such as long running cryptographic computations. The use of checkpoints to ensure program progress despite unreliable power is a common solution for the availability of intermittent systems [1], [2] and one that has even gathered support from device manufacturers [3]. These developments have been enabled by the growth of write-efficient non-volatile memory, such as ferroelectric RAM (FRAM), and its availability for use in modern microcontrollers. FRAM, and similar memories, provide an opportunity to read and write from non-volatile memory at nearly the same speed and cost of standard SRAM, allowing fast, permanent, storage of critical system data and easy resumption of system state following power loss.

The result of these innovations are intermittent systems that can continue operation even when power is frequently interrupted before a normal operation would complete. However, very little research has explored the security implications of intermittent systems or acknowledged the capabilities of an adversary who can interrupt power to an intermittent device. The majority of embedded system security research is not directed at intermittent systems, incorrectly assuming that power loss or reset will return the system to a known, safe, initial state, and explicitly states that an adversary cannot have any effect on the physical system itself to meet their security guarantees [4]. This is an unreasonable assumption for an attacker targeting systems specifically designed to operate on unreliable power and potentially remote environments, and our demonstration will illustrate a first step toward addressing the security concerns of intermittent systems.

Department of Electrical and Computer Engineering
Virginia Tech, Blacksburg, VA 24061 USA
Email: {archanaa, cesuslow, ddinu, schaum}@vt.edu

Our research considered these security implications and presented a secure checkpointing technique, the Secure Intermittent Computing Protocol (SICP), that is effective against an attacker that may attempt to scan, alter or replay sensitive program state stored in system checkpoints.

We argue that three properties must be present in a secure checkpointing system:

1) Each checkpoint must be tied to a unique power-on state such that it can be restored only once.
2) The sequence of checkpoints must be securely maintained to ensure only the most recent checkpoint may be restored.
3) The protocol itself must be resilient to power loss during its operations.

Our protocol satisfies these requirements and has been implemented on an MSP430FR5994 microcontroller to test its effectiveness. We are able to ensure that each system checkpoint is unique through the introduction of fresh random values from a true random number generator at each system restart. The checkpoints are cryptographically chained together to preserve their order and authenticity, preventing an adversary from loading a checkpoint from another device onto a system to produce specific behavior nor restoring a system to its own checkpoint that is not the most recent. Finally, the system itself is robust against power loss during its operation preventing corruption of checkpoints if power is lost again before complete restoration of the state.

To our knowledge this is the first work to address these security concerns of intermittent systems and present a solution to provide secure checkpointing functionality to intermittent systems.

II. Demonstration Description

Our hardware demonstration will illustrate the features of SICP and highlight the potential for properly securing intermittent systems. To do this, we will provide the audience with complete control over the power supply to the device and insight into the system's current operation to observe the effects of power loss on the current execution of the device.

The demonstration will provide an opportunity for the audience to explore the capabilities of an attacker able to manipulate the power of an intermittent system and verify the effectiveness of SICP at preserving correct execution of the loaded program.

III. Experimental Setup

We demonstrate our system's capabilities by providing the audience (our attacker) control of our power supply via a

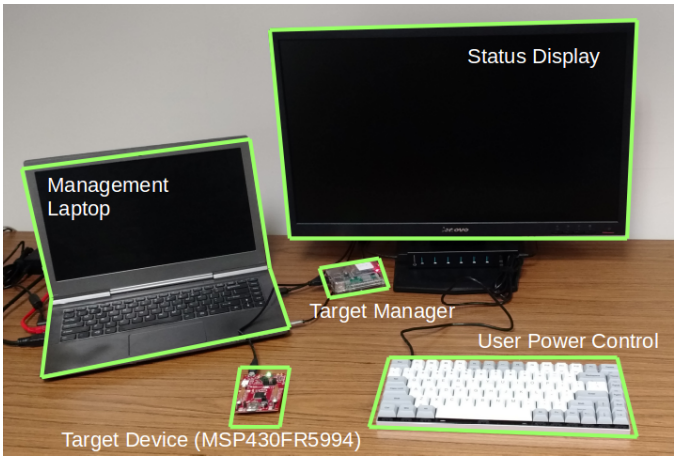

Fig. 1. Demonstration layout with components. The user power control will directly manage the power to the target device during the demonstration.

single large button. The state of the system is displayed on a connected computer system. Th display includes the cryptographic hashes of the current and previous checkpoints, the cryptographic tags associated with each state, the progress of the current software computation, and current security mode.

This construction allows the audience to observe the following about the system:

1) *Normal Operation*: Without unexpected interruption the system will execute continued software based AES encryptions with progress shown for the completion of each AES round. This will allow the audience to see the normal operation of the system, when checkpoints are taken, and the expected behavior when reliable power is present.
2) *Checkpoint Restoration*: When power is removed, the audience will be able to observe most recent checkpoint, its authentication and restoration by SICP, and the system's resumption of its previous operation from the most recent checkpoint location.
3) *Uniqueness of Checkpoints*: It is impossible to generate the same tag, or hash, even if the system is at the same point in its operation. This shows the uniqueness created for each checkpoint and demonstrates the impossibility of playing back the same state a second time.
4) *Protocol Robustness Against Power Loss*: If power is interrupted during SICP operations, it will not negatively affect the performance of the system or its security.
5) *Sanitation of State Information*: Sensitive data that accumulates between checkpoints is wiped from the system when power is removed. This is shown through a snap shot of the secure memory section shown to the user via our demonstration interface. The audience will be able to verify that an adversary would not be able to observe sensitive data if they examine memory after power is removed.

After viewing the demonstration, the value of employing a secure checkpointing system for intermittent computing will be clear to observers. The audience will have had an opportunity to exploit an attack vector that is normally excluded from the attacker model of embedded security solutions, control of the device's power supply, and they will have seen that the system cannot be forced into an invalid or weakened state with this adversary capability. Ultimately, the demonstration will have provided an opportunity to discuss the security challenges facing intermittent systems and how they can be addressed as the deployment of intermittent systems grows in the future.

A. Equipment List

To support this demonstration we will use the following equipment:

- *Display monitor (24")*: audience feedback for system state.
- *Laptop*: developer interface and serial connection to target device.
- *MSP430FR5994 Development Board*: target device implementing SICP.
- *User Button*: audience control of power supply for target device.

B. Organization

Fig. 1 shows the organization of the demonstration. The audience is presented with a button to control power to the device and invited to interrupt the ongoing computation. The display monitor shows the current state of the system including progress of the computation, current and previous checkpoint statistics, expected values, the current data in secure non-volatile memory, and a hash of the current system state information.

The audience feedback will help illustrate the operation of SICP and how its components interact to secure the intermittent system under test. The display will also allow an observer to verify that it is not possible to replay an identical system state through manipulation of the system's power supply. Finally, the observer will be able to validate through the demonstration that there is no point in the system's operation where an attacker's control of the power supply would break the checkpoint security or result in sensitive state data being improperly left in non-volatile memory.

References

[1] H. Jayakumar, A. Raha, and V. Raghunathan, "Quickrecall: A low overhead hw/sw approach for enabling computations across power cycles in transiently powered computers," in *2014 27th International Conference on VLSI Design and 2014 13th International Conference on Embedded Systems*, pp. 330–335, Jan 2014.

[2] B. Lucia and B. Ransford, "A simpler, safer programming and execution model for intermittent systems," in *Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI '15, (New York, NY, USA), pp. 575–585, ACM, 2015. DINO.

[3] Texas Instruments, *MSP MCU FRAM Utilities*, 2017.

[4] J. Noorman, P. Agten, W. Daniels, R. Strackx, A. Van Herrewege, C. Huygens, B. Preneel, I. Verbauwhede, and F. Piessens, "Sancus: Low-cost trustworthy extensible networked devices with a zero-software trusted computing base," in *Proceedings of the 22Nd USENIX Conference on Security*, SEC'13, (Berkeley, CA, USA), pp. 479–494, USENIX Association, 2013.