TARDIS: Time and Remanence Decay in SRAM to Implement Secure Protocols on Embedded Devices without Clocks

Amir Rahmati
UMass Amherst

Mastooreh Salajegheh
UMass Amherst

Dan Holcomb
UC Berkeley

Jacob Sorber
Dartmouth College

Wayne P. Burleson
UMass Amherst

Kevin Fu
UMass Amherst

Abstract

Lack of a locally trustworthy clock makes security protocols challenging to implement on batteryless embedded devices such as contact smartcards, contactless smartcards, and RFID tags. A device that knows how much time has elapsed between queries from an untrusted reader could better protect against attacks that depend on the existence of a rate-unlimited encryption oracle.

The TARDIS (Time and Remanence Decay in SRAM) helps locally maintain a sense of time elapsed without power and without special-purpose hardware. The TARDIS software computes the expiration state of a timer by analyzing the decay of existing on-chip SRAM. The TARDIS enables coarse-grained, hourglass-like timers such that cryptographic software can more deliberately decide how to throttle its response rate. Our experiments demonstrate that the TARDIS can measure time ranging from seconds to several hours depending on hardware parameters. Key challenges to implementing a practical TARDIS include compensating for temperature and handling variation across hardware.

Our contributions are (1) the algorithmic building blocks for computing elapsed time from SRAM decay; (2) characterizing TARDIS behavior under different temperatures, capacitors, SRAM sizes, and chips; and (3) three proof-of-concept implementations that use the TARDIS to enable privacy-preserving RFID tags, to deter double swiping of contactless credit cards, and to increase the difficulty of brute-force attacks against e-passports.

1 Introduction

> "Timestamps require a secure and accurate system clock—not a trivial problem in itself."
> –Bruce Schneier, *Applied Cryptography* [43]

Even a perfect cryptographic protocol can fail without a trustworthy source of time. The notion of a trustworthy clock is so fundamental that security protocols rarely state this assumption. While a continuously powered computer can maintain a reasonably accurate clock without trusting a third party, a batteryless device has no such luxury. Contact smartcards, contactless smartcards, and RFIDs can maintain a locally secured clock during the short duration of a power-up (e.g., 300 ms), but not after the untrusted external reader removes power.

Platform	Attack	#Queries
MIFARE Classic	Brute-force [15]	$\geq$1,500
MIFARE DESFire	Side-channel [35]	250,000
UHF RFID tags	Side-channel [34]	200
TI DST	Reverse eng. [7, 8]	$\sim$75,000
GSM SIM card	Brute-force [16]	150,000

Table 1: Practical attacks on intermittently powered devices. These attacks require repeated interactions between the reader and the device. Throttling the reader's attempts to query the device could mitigate the attacks.

It's Groundhog Day! Again. Unawareness of time has left contactless payment cards vulnerable to a number of successful attacks (Table 1). For instance, Kasper et al. [35] recently demonstrated how to extract the 112-bit key from a MIFARE DESFire contactless smartcard (used by the Clipper all-in-one transit payment card[1]). The side channel attack required approximately 10 queries/s for 7 hours. Some RFID credit cards are vulnerable to replay attacks because they lack a notion of time [21]. Oren and Shamir [34] show that power analysis attacks on UHF RFID tags can recover the password protecting a "kill" command with only 200 queries. At USENIX Security 2005, Bono et al. [8] implemented a brute-force attack against the Texas Instruments Digital Signature Transponder (DST) used in engine immobilizers and the ExxonMobile SpeedPass™. The first stage of the attack required approximately 75,000 online "oracle" queries to

[1] No relation to the Clipper Chip [27].

Accepted to appear at the 21st USENIX Security Symposium, August 2012.

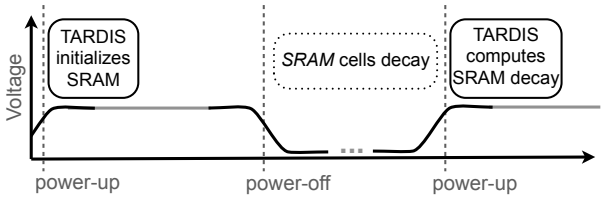

Figure 1: TARDIS estimates time by counting the number of SRAM cells that have a value of zero in power-up (*computes SRAM decay*). Initially, a portion of SRAM cells are set to one (*initializes SRAM*) and their values decay during power-off. The dots in the power-off indicate the arbitrary and unpredictable duration of power-off.

recover the proprietary cipher parameters [7].

A batteryless device could mitigate the risks of brute-force attacks, side-channel attacks, and reverse engineering by throttling its query response rate. However, the tag has no access to a trustworthy clock to implement throttling. A smartcard does not know whether the last interrogation was 5 seconds ago or 5 days ago.

Enter the TARDIS. To enable security protocols on intermittently powered devices without clocks, we propose Time and Remanence Decay in SRAM (TARDIS) to keep track of time without a power source and without additional circuitry. The TARDIS relies on the behavior of decaying SRAM circuits to estimate the duration of a power failure (Figure 1). Upon power-up, the TARDIS initializes a region in SRAM of an intermittently powered device. Later, during power-off, the SRAM starts to decay. Upon the next power-up, the TARDIS measures the fraction of SRAM cells that retain their state. In many ways, TARDIS operation resembles the functioning of an hourglass: the un-decayed, decaying, and fully decayed stages of SRAM are analogous to full, emptying, and empty hourglass states.

Contributions. Our primary contributions are:

- Algorithmic building blocks to demonstrate the feasibility of using SRAM for a trustworthy source of time without power.
- Empirical evaluation that characterizes the behavior of SRAM-based timekeeping under the effects of temperature, capacitance, and SRAM size.
- Enabling three security applications using SRAM-based TARDIS: sleepy RFID tags, squealing credit cards, and forgiving e-passports.

State of the Art. Today, batteryless devices often implement monotonically increasing counters as a proxy for timekeeping. RFID credit cards occasionally include transaction counters to defend against replay attacks. Yet the counters introduce vulnerabilities for denial of service and are difficult to reset based on time elapsed; one credit card ceases to function after the counter rolls over [21]. While one can maintain a real-time clock (RTC) with a battery on low-power mobile devices [40], batteryless platforms do not support RTCs across power failures [31, 41, 9] because of the quiescent current draw.

While a timer of just a few seconds would suffice to increase the difficulty of brute-force attacks (Table 1), our experimental results indicate that an SRAM timer can reliably estimate the time of power failures from a few seconds up to several hours. For example, using a 100 μF capacitor at room temperature, the TARDIS expiration time can exceed 2 hours of time. We evaluate the energy and time overhead of the TARDIS, its security against thermal and power-up attacks, and its precision across different platforms.

The primary novelty of the TARDIS is that a moderately simple software update can enable a sought-after security primitive on existing hardware without power. While data remanence is historically considered an undesirable security property [19], the TARDIS uses remanence to improve security. At the heart of the TARDIS are SRAM cells, which are among the most common building blocks of digital systems. The ubiquity of SRAM is due in part to ease of integration: in contrast with flash memory and DRAM, SRAM requires only a simple CMOS process and nominal supply voltage.

2 Intermittently Powered Devices: Background, Observations, and Challenges

New mobile applications with strict size and cost constraints, as well as recent advances in low-power microcontrollers, have given rise to a new class of intermittently powered device that is batteryless and operates purely on harvested energy. These devices—including contact and contactless smart cards and computational RFID tags (CRFIDs) [38, 41, 56, 55]— typically have limited computational power, rely on wireless transmissions from a reader both for energy and for timing information, and lose power frequently due to minimal energy storage. For example, when a contactless transit card is brought sufficiently close to a reader in a subway, the card gets enough energy to perform the requested tasks. As soon as the card is out of the reader range, it loses power and is unable to operate until presented to another reader. Since a tag loses power in the absence of a reader, it doesn't have any estimation of time between two interactions with a reader.

A typical secure communication between a reader and a tag is shown in Figure 2. The tag will only respond to the reader's request if it has authenticated itself by correctly answering the challenge sent by the tag. Two problems

Accepted to appear at the 21st USENIX Security Symposium, August 2012.

	SRAM	DRAM
Purpose	Fast local memory	Large main memory
Location	Usually on-chip w/ CPU	Usually off-chip
Applications	CPU caches, microcontrollers	Desktop computers, notebooks, servers
Storage technology	Cross-coupled transistors	Capacitors
Normal operation	Constantly powered	Intermittently refreshed
Decay state	50% zero/one bits	All zero bits

Table 2: Because CPUs of embedded devices generally do not have on-chip DRAM, the TARDIS operates on SRAM. SRAM and DRAM differ fundamentally in their manufacture, operation, intended use, and state of decay.

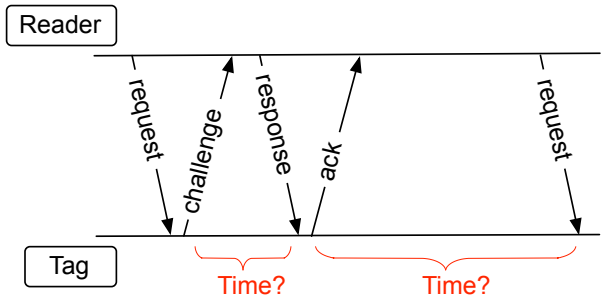

Figure 2: The tag cannot determine the time between a challenge and a response or the time between two sessions. The reader could respond to the tag as tardily as it likes or query the tag as quickly as it wants.

arise in this scheme:

- The tag is unaware of the amount of time spent by the reader to answer the challenge, so an adversary has an unlimited amount of time to crack a challenge.
- The tag is unaware of the time between two different queries, so an adversary can send a large number of queries to the tag in a short time space. This can make various brute-force attacks possible on these devices.

Traditionally, computing devices have either had a direct connection to a reliable power supply or large batteries that mask disconnections and maintain a constant supply of power to the circuit. In either case, a reliable sense of time can be provided using an internal clock. Time measurement errors, due to clock drift or power failures, can be corrected by synchronizing with a trusted peer or other networked time source. Current embedded systems address the timekeeping issue in one of the following ways:

1. A system can power a real-time clock (RTC); however, this is not practical on intermittently powered devices due to their tight energy budget. Even if the system uses a low-power RTC (e.g., NXP PCF2123 RTC chip [32]), the RTC component has to be constantly powered (for example, using a battery). This choice also increases the cost of manufacturing and it does not benefit devices that are already deployed.
2. A system can keep time by accessing an external device (e.g., an RFID tag reader) or by secure time synchronization [14, 46]. This option introduces security concerns and may either require significant infrastructure or severely limit range and mobility.

2.1 Threat Model and Assumptions

> "...if the attack surface includes an awful lot of clocks that you do not control, then it's worth some effort to try and make your system not depend on them anymore."–Ross Anderson [30]

The primary goal of the adversary in our model is to distort the TARDIS timekeeping. Our threat model considers semi-invasive attacks common to smart cards [15, 35]. We will not discuss attacks such as buffer overflows which are against the systems that would integrate the TARDIS; we focus on the attacks aimed at the TARDIS itself. Our adversarial model considers two classes of attacks: (1) thermal attacks that use heating and cooling [19] to distort the speed of memory decay; and (2) power-up attacks that keep the tag partially powered to prevent memory decay.

3 The TARDIS Algorithms

The TARDIS exploits SRAM decay during a power-off to estimate time. An example of the effect of time on SRAM decay in the absence of power is visualized in Figure 3. In this experiment, a 100×135 pixel bitmap image of a different TARDIS [1] was stored into the SRAM of a TI MSP430 microcontroller. The contents of the memory were read 150, 190, and 210 seconds after the power was disconnected. The degree of image distortion is a function of the duration of power failure.[2]

[2]The 14.6 KB image was too large to fit in memory, and therefore was divided into four pieces with the experiment repeated for each to

Accepted to appear at the 21st USENIX Security Symposium, August 2012.

Figure 1 shows the general mechanism of the TARDIS. When a tag is powered up, the TARDIS initializes a region in SRAM cells to 1. Once the power is cut off, the SRAM cells decay and their value might reset from 1 to 0. The next time the tag is powered up, the TARDIS tracks the time elapsed after the power loss based on the percentage of cells remaining 1. Algorithm 1 gives more details about the implementation of the TARDIS.

MEASURE_TEMPERATURE: To detect and compensate for temperature changes that could affect the decay rate (Section 6), the TARDIS uses the on-board temperature sensor found on most microcontrollers. The procedure MEASURE_TEMPERATURE stores inside-the-chip temperature in the flash memory upon power-up. The procedure DECAY calls the TEMPERATURE_ANALYZE function to decide if the temperature changes are normal.

TIME: The TARDIS TIME procedure returns *time* and *decay*. The precision of the *time* returned can be derived from the *decay*. If the memory decay has not started ($decay = 0$), the procedure returns $\{time, 0\}$ meaning that the time duration is less than *time*. If the SRAM decay has started but has not finished yet ($0 \leq decay \leq 50\%$), the return value *time* is an estimate of the elapsed time based on the *decay*. If the SRAM decay has finished ($decay \simeq 50\%$), the return result is $\{time, 50\}$ meaning that the time elapsed is greater than *time*.

ESTIMATION: The procedure ESTIMATE uses a lookup table filled with entries of decay, temperature, and time stored in non-volatile memory. This table is computed based on a set of experiments on SRAM in different temperatures. Once the time is looked up based on the measured decay and the current temperature, the result is returned as *time* by the ESTIMATE procedure. The precompiled lookup table does not necessarily need to be calibrated for each chip as we have observed that chip-to-chip variation affects decay only negligibly (Section 6).

Algorithm 1 TARDIS Implementation

```
INIT(addr, size)
1  for i ← 1 to size
2       do memory(addr + i − 1) ← 0xFF
3  temperature ← MEASURE_TEMPERATURE()

DECAY(addr, size)
1  decay ← COUNT0S(addr, size)
2  ≻ Proc. COUNT0S counts the number of 0s in a byte.
3  if TEMPERATURE_ANALYZE(temperature)
4  ≻ This procedure decides if the temperature changes
     are expected considering the history of temperature
     values stored in flash memory.
5     then return decay
6     else return error

EXPIRED(addr, size)
1  ≻ Checks whether SRAM decay has finished.
2  decay ← DECAY(addr,size)
3  if (decay ≥ %50 × 8 × size)
4     then return true
5     else return false

TIME(addr, size, temperature)
1  ≻ Estimate the passage of time by comparing the
     percentage of decayed bits to a precompiled table.
2  decay ← DECAY(addr,size)/(8 × size)
3  time ← ESTIMATE(decay,temperature)
4  return {time, decay}
```

3.1 TARDIS Performance

The two most resource-consuming procedures of the TARDIS are INIT (initializing parts of the SRAM as well as measuring and storing the temperature) and DECAY (counting the zero bits and measuring the temperature). Table 3 shows that energy consumed in total by these two procedures is about 48.75 μJ and it runs in 15.20 ms.

Our experiments of time and energy measurements are performed on Moo RFID[56] sensor tags that use an MSP430F2618 microcontroller with 8 KB of memory, and a 10 μF capacitor. A tag is programmed to perform one of the procedures, and the start and end of the task is marked by toggling a GPIO pin. The tag's capacitor is charged up to 4.5 V using a DC power supply and then disconnected from the power supply so that the capacitor is the only power source for the tag. In the experiments, the DC power supply is used instead of an RF energy supply because it is difficult to disconnect the power harvesting at a precise capacitor voltage. We measured the voltage drop of the capacitor and the GPIO pin toggling using an oscilloscope. The energy consumption of the task is the difference of energy ($\frac{1}{2} \times CV^2$) at the start and end of the task. The reported measurement is the average of ten trials.

get the complete image. The microcontroller was tested in a circuit shown in Figure 6 with a 10 μF capacitor at 26°C. No block transfer computation was necessary.

4 Securing Protocols with the TARDIS

There are many cases where the security of real-world applications has been broken because the adversary could query the device as many times as required for attack. Table 1 gives a summary of today's practical attacks on intermittently powered devices. By integrating the TARDIS, these applications could throttle their response rates and

Accepted to appear at the 21st USENIX Security Symposium, August 2012.

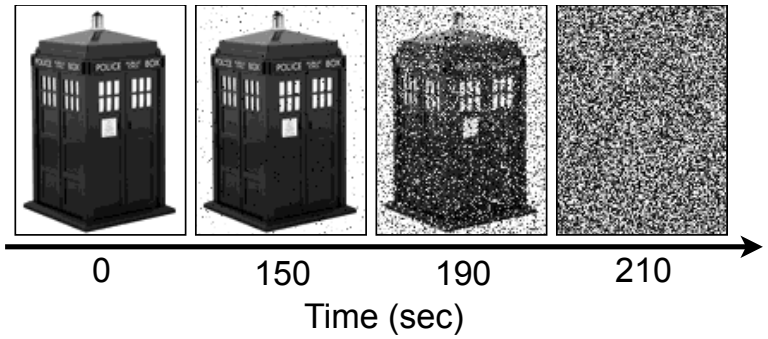

Figure 3: Programs without access to a trustworthy clock can determine time elapsed during a power failure by observing the contents of uninitialized SRAM. These bitmap images of the TARDIS [1] represent four separate trials of storing the bitmap in SRAM, creating an open circuit across the voltage supply for the specified time at $26°C$, then immediately returning a normal voltage supply and reading uninitialized SRAM upon reboot. The architecture of a contactless card is modeled using a 10 μF capacitor and a diode in series with the MSP430 microcontroller's voltage supply pin. The degree of decay is a function of the duration of power failure, enabling hourglass-like timekeeping precision without power. No TARDIS was harmed or dematerialized in this experiment.

Procedure	**Energy Cost**	**Exec. Time**
INIT	$11.53\ \mu J \pm 2.47$	$2.80\ ms \pm 0.0\bar{0}$
DECAY	$37.22\ \mu J \pm 9.31$	$12.40\ ms \pm 1.10$

Table 3: Overhead of TARDIS INIT and DECAY procedures measured for TARDIS *size* of 256 bytes.

improve their security.

We discuss six security protocols that could strengthen their defense against brute-force attacks by using the TARDIS. To demonstrate the ease of integrating the TARDIS, we have implemented and tested three of these security protocols on the Moo, a batteryless microcontroller-based RFID tag with sensors but without a clock [56]. Our prototypes demonstrate the feasibility of the TARDIS and its capabilities in practice.

Sleepy RFID Tags: To preserve the users privacy and prevent traceability, one could use a "kill" command to permanently deactivate RFID tags on purchased items [25]. However, killing a tag disables many features that a customer could benefit from after purchase. For example, smart home appliances (e.g., refrigerators or washing machines) may no longer interact with related items even though they have RFID tags in them. One could temporarily deactivate RFID tags by putting them to "sleep." However, lack of a simple and practical method to wake up the tags has made this solution inconvenient [25]. By providing a secure notion of time, the TARDIS makes it possible to implement *sleepy tags* that can sleep temporarily without requiring additional key PINs or cryptographic solutions. We consider a time resolution on the order of hours more appropriate for this application.

To extend the sleep time of sleepy tags, one could use a counter along with the TARDIS as follows: upon power-up, the tag checks the TARDIS timer, and it does not respond to the reader if the timer has not expired. If the TARDIS timer has expired, the tag decreases the counter by one and initializes the TARDIS again. This loop will continue while the counter is not zero. For example, using a counter initially set to 1000 and a TARDIS resolution time of 10 seconds, the tag could maintain more than 2 hours of delay. Since the tag exhausts its counter every time it wakes up, the reader interacting with the tag has to query the tag intermittently.

The TARDIS could prevent yet another attack on Electronic Product Code (EPC) tags that use "kill" commands. To prevent accidental deactivation of tags, a reader must issue the right PIN to kill a tag [12]. An adversary could brute-force the PIN (32 bits for EPC Class1 Gen2 tags). The TARDIS enables the RFID tag to slow down the unauthorized killing of a tag by increasing the delay between queries and responses.

Squealing Credit Cards: Today, a consumer cannot determine if her card has been used more than once in a short period of time unless she receives a receipt. This is because a card cannot determine the time elapsed between two reads as the card is powered on only when it communicates with the reader. The TARDIS enables a "time lock" on the card such that additional reads would be noticed. Thus a consumer could have some assurance that after exposing a card to make a purchase, an accidental second read or an adversary trying to trick the card into

Accepted to appear at the 21st USENIX Security Symposium, August 2012.

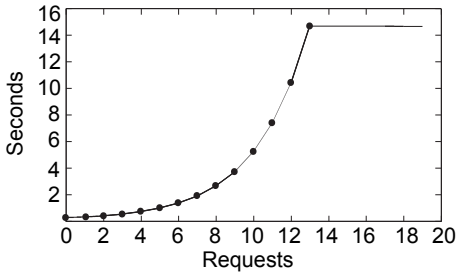

Figure 4: Measured response time of a 2010-issued French passport [5]. The passport imposes up to 14 seconds of delay on its responses after unsuccessful execution. The delay will remain until a correct reading happens even if the passport were removed from the reader's field for a long time.

responding would be revealed. *Squealing credit cards* would work similarly to today's credit cards, but they are empowered by the TARDIS to estimate the time between queries and warn the user audibly (a cloister bell) if a second read is issued to the card too quickly. A time lock of about one minute can be considered enough for these applications.

Forgiving E-passports: RFID tags are used in e-passports to store holder's data such as name, date of birth, biometric ID, and a unique chip ID number. E-passports are protected with techniques such as the Basic Access Control (BAC) protocol, shielding, and passive authentication. However, in practice, e-passports are not fully protected. An adversary can brute-force the BAC key in real time by querying the passport 400 times per minute for a few weeks [6]. Another attack can accurately trace a specific passport by sending hundreds of queries per minute [11].

To mitigate the effect of brute-force attacks, French e-passports have implemented a delay mechanism—we imagine using a counter—to throttle the read rate [5]. This delay increases to 14 seconds after 14 unsuccessful attempts (Figure 4) and would occur even if the passport was removed from the RF field for several days. Once the tag is presented with an authorized reader, the delay will be enforced and then reset to zero. The TARDIS provides a time-aware alternative that delays unauthorized access but ignores the previous false authentication attempts if the passport has been removed from the reader's range for an appropriate duration. A time duration matching the maximum implemented delay (14 seconds for French passports) would be enough to implement this function.

Passback - Double-tap Prevention: In mass transportation and other similar card entry systems, the goal of the operator is to prevent multiple people from accessing the system simultaneously using the same card. To achieve this goal, systems are typically connected to a central database that prevents a card from being used twice in a short time frame.[3] Using the TARDIS, a card could implement delay before permitting re-entry rather than requiring the system to check a central database.

Resurrecting Duckling: Secure communication in ad-hoc wireless networks faces many obstacles because of the low computing power and scarce energy resources of these devices. Stajano et al. [45] proposed a policy in which these devices would transiently accept a new owner. The devices will later return to an unprogrammed status when the owner no longer needs them, they receive a kill command, or another predefined reset condition is met. Later, others can reclaim and reuse these devices.

For wirelessly powered devices, the TARDIS can provide a sense of time, allowing them to be "reborn" with a new owner only if there is an extended power outage. A legitimate user can continue to power the device wirelessly, but if she wishes to transfer ownership to another entity, she must power it down for a long enough time (defined by the user). Otherwise, the RFID tag refuses to interact with anyone not possessing the present cryptographic key. An example of this application is secure pairing for computational contact lenses [22]. The controller could be cryptographically bound until power disappears for more than a few minutes. Another use of this application is to make stealing SIM cards difficult [16]. The card could refuse to boot if it has been unpowered for a fair amount of time.

Time-out in Authentication Protocols: Because RFID tags rely on a reader as their source of energy, they cannot measure the delay between a request to the reader and its corresponding response. The tag ignorance gives the reader virtually unlimited time to process the request and response in an authentication algorithm. Having unlimited response time enables the adversary to employ various attacks on the request message with the goal of breaking it. Using the TARDIS will limit the adversary time frame for a successful attack. An example of these protocols can be seen in the e-passport BAC protocol where the reader and passport create a session key for communication. Using The TARDIS would enable passports to enforce expiration of these keys.

4.1 Implementation and Evaluation

For the implementation of *sleepy tags*, *squealing credit cards*, and *forgiving e-passports*, we have chosen the Moo, a batteryless microcontroller-based RFID tag. We have

[3] Houston METRO system: `http://www.ridemetro.org/fareinfo/default.aspx`

Accepted to appear at the 21st USENIX Security Symposium, August 2012.

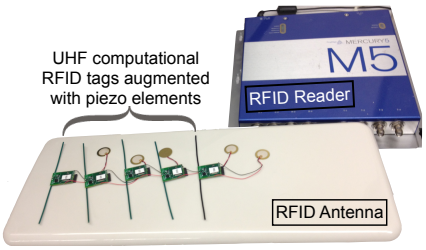

Figure 5: Our applications are implemented and tested on the Moo RFID sensors and are remotely powered by a RFID reader (ThingMagic M5 [51]).

Algorithm 2 An example of TARDIS usage in a protocol.

```
TARDIS_EXAMPLE(addr,size)
1  if EXPIRED(addr,size)
2     then RESPOND_TO_READER()
3          INIT(addr,size)
4     else BUZZ_PIEZO_ELEMENT()
```

augmented this tag with a piezo-element [20] so that it can audibly alert the user to events.

Implementation: We have implemented a TARDIS library that provides the procedures INIT and EXPIRE listed in Algorithm 1. For the three implemented protocols, a 1-bit precision of time–whether or not the timer had expired–was enough. The programs used for all three protocols are similar and are shown in Algorithm 2. The tag was programmed to call the EXPIRE procedure upon power-up; if the timer had expired, it would respond to the reader and call INIT; otherwise, the tag would buzz its piezo-element. In the case of the *squealing credit cards* protocol the tag was programmed to respond to the reader after buzzing, but for the two other applications, the tag stopped communicating with the reader.

We used a ThingMagic reader [51] and its corresponding antenna to query the tag. When the tag was queried for the first time upon removal from the RF field, it buzzed. The tag stayed quiet whenever it was queried constantly or too quickly.

Experimental Setup: To measure the TARDIS resolution time on this platform, we powered up the tag to 3.0 V using an external power supply and then disconnected it. We observed the voltage drop over time on an oscilloscope and measured the elapsed time between loss of power and when SRAM decay has finished.[4] We conducted our experiments on five tags, which use a 10 μF capacitor as its primary power source. The TARDIS resolution time on average was 12.03 seconds with a standard deviation of 0.11 seconds. A similar tag, which uses 100 mF, yields a TARDIS resolution time of 145.85 seconds. These time measurements are specific to the platform we have chosen for our experiment. The resolution could potentially be extended to hours using additional capacitors (Table 5).

[4]Our experiments (Section 6) have shown that SRAM decay finishes when the tag voltage reaches 50 mV.

5 Security Analysis

Depending on the application, the adversary may wish either to slow down or to speed up the expiration of the TARDIS. We discuss four different attacks that try to distort the TARDIS interpretation of time.

Cooling Attacks. An adversary might try to reduce the system's temperature, aiming to slow down the memory decay rate. Other works [19] have used this technique to prevent data decay in DRAM for the purpose of data extraction. Cooling attacks might target the TARDIS timer in cases where the adversary needs to slow the passage of time. As explained in Algorithm 1, the TARDIS measures and records a device's temperature over time and therefore it can prevent cooling attacks by observing unexpected temperature changes.

Heating Attacks. In contrast to cooling attacks, an attacker might need to speed up the TARDIS timer. For example, someone might try to decrease the delay between queries in order to speed up brute-force attacks. Similarly to the defense against cooling attacks, the TARDIS will report an error indicating unexpected temperature changes.

Pulse Attacks. A more sophisticated attack is a combination of the cooling and heating attacks such that the temperature would remain the same in the beginning and the end of the attack. It should be noted that this is not a trivial attack because the adversary needs to restore the original internal temperature to prevent the thermal sensor from noticing any difference. A defense against pulse attacks is to implement a thermal fuse [10] on the chip that will activate when the chip is exposed to a high temperature. The activation of this fuse will then either notify the TARDIS of temperature tampering on the next boot-up or possibly prevent the system from booting up at all.

Voltage Control Attack. Another possible attack scenario would be to power up the system wirelessly to a minimum voltage that is not sufficient for booting up but sufficient for stopping the memory decay. This would prevent the device from noticing the unauthorized reader and it would stop the memory from decaying further (see Figure 8). The voltage control attack can freeze the TARDIS timer at a specific time as long as it sustains the power sup-

Accepted to appear at the 21st USENIX Security Symposium, August 2012.

ply. We imagine that this attack is difficult to implement because of the inherent design of the readers. Many factors (e.g., distance) affect the voltage received by the tags and tags are very sensitive to environmental effects. The readers are also generally designed to flood the targeted environment with energy to provide the tags in range with more than the maximum required power [54]. Excessive power that may have been generated by these devices is then filtered out in tags using voltage regulators. To implement this attack, we imagine the adversary would need to control the input voltage to the tag with a very high precision. If the tag voltage for any reason drops, the SRAM will decay irreversibly. At the same time, the adversary would need to prevent the tags from fully powering up and noticing the unauthorized reader.

6 Factors Affecting SRAM Decay

In our evaluation of the TARDIS, we examine the decay behavior of SRAM and three factors that have major effects on this behavior. All experiments use the same circuit (Figure 6), and follow the same general procedure.

Experimental Setup: A microcontroller runs a program that sets all available memory bits to 1. The power is then effectively disconnected for a fixed amount of time (*off-time*). When power is reapplied to the chip, the program records the percentage of remaining 1-bits to measure memory decay, and then it resets all bits to 1 in preparation for the next time power is disconnected. A Data Acquisition (DAQ) unit from Agilent (U2541A series) precisely controls the timing of power-ups and power-downs between 3 and 0 Volts, and also measures the voltage across the microcontroller throughout the experiment. An inline diode between the power supply and microcontroller models the diode at the output of the power harvesting circuit in RFIDs; it also prevents the DAQ from grounding VCC during the off-time when the DAQ is still physically connected but is not supplying power. In all experiments, microcontrollers from the TI MSP430 family are used to ensure maximum consistency. The microcontroller used in all experiments is MSP430F2131 with 256 B of SRAM unless stated otherwise.

In all of the experiments, temperature is controlled by conducting all tests inside of a Sun Electronics EC12 Environmental Chamber [47] capable of creating a thermally stable environment from $-184°C$ to $+315°C$ with $0.5°C$ precision. We use an OSXL450 infrared non-contact thermometer [33] with $\pm 2°C$ accuracy to verify that our microcontroller has reached thermal equilibrium within the chamber before testing. For all the experiments, we have collected at least 10 trials.

Defining Stages of Decay: Three distinct stages of decay are observed in all experiments. Figure 7 illustrates the three stages of SRAM decay measured on a TI MSP430F2131 with 256 B of SRAM and a 10 μF capacitor, at $26°C$. We vary the *off-time* from 0 to 400 seconds in 20-second increments. In the first stage, no memory cells have decayed; during the second stage, a fraction of the cells, but not all, have decayed; by the third stage the cells have decayed completely (see Table 4 for a summary of term definitions). Observations made during Stages 1 or 3 provide a single bit of coarse information, indicating only that Stage 2 has not yet begun or else that Stage 2 has already been completed. Observations made during Stage 2 can provide a more accurate notion of time based on the percentage of decayed bits.

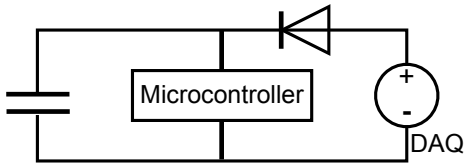

Figure 6: General circuit used during the experiments. The microcontroller is held in an environmental chamber to ensure consistent temperature during the tests. The Data Acquisition (DAQ) unit both provides power to the microcontroller and records the voltage decay.

Term	Definition
SRAM Decay	Change of value in SRAM cells because of power outage
Decay Stage 1	Time before the first SRAM cell decays
Decay Stage 2	Time between the decay of first SRAM cell and last one
Decay Stage 3	Time after the last SRAM cell decays
Ground State	The state that will be observed in an SRAM cell upon power-up, after a very long time without power
DRV	Data Retention Voltage, minimum voltage at which each cell can store a datum
DRV Probability(v)	Probability that a randomly chosen cell will have a DRV equal to v and a written state that is opposite its ground state.

Table 4: Definition of the terms used to explain the behavior of SRAM decay and the theory behind it.

Decay vs. Voltage: The decay rate of SRAM is expected to depend only on its voltage level (Section 7). Temperature, SRAM size, and circuit capacitance all affect the rate of voltage depletion and thus only have secondary effects on memory decay. Our experimental results (Figure 8) for five sets of tests (each at least 10 trials) support this hypothesis. The same setup as explained before was

Accepted to appear at the 21st USENIX Security Symposium, August 2012.

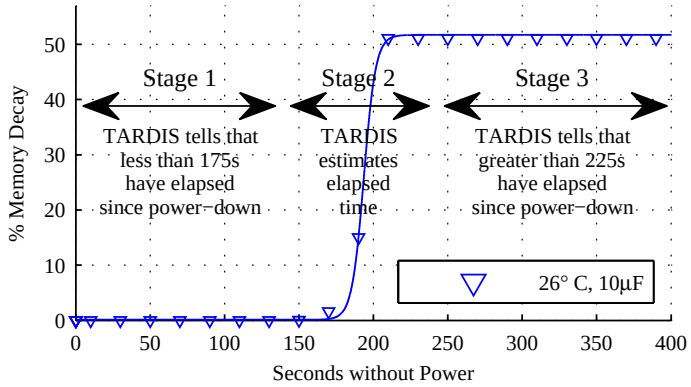

Figure 7: The TARDIS presents a three-stage response pattern according to its amount of decay. Before 175 seconds, the percentage of bits that retain their 1-value across a power-off is 100%. For times exceeding 225 seconds, the TARDIS memory has fully decayed. The decay of memory cells between these two thresholds can provide us with a more accurate measurement of time during that period. This graph presents our results measured on a TI MSP430F2131 with 256 B of SRAM and a 10 μF capacitor at 26°C.

used and five different temperatures (one with a 10 mF capacitor and four of them without) were tested.

Impact of Temperature: The work of Skorobogatov [44] shows that low temperature can increase the remanence time of SRAM, and the work of Halderman et al. [19] similarly shows that low temperature can extend the remanence time of DRAM. For the TARDIS using SRAM decay to provide a notion of time, the interesting question is the opposite case of whether high temperature can decrease remanence. We use the same experimental setup as before (without using capacitors) to investigate how decay time varies across five different elevated temperatures (in the range of $28°C - 50°C$). The off-time of the microcontroller varied from 0 to a maximum of 5 seconds. Figure 9 shows that the decay time is non-zero across all temperatures. This indicates that the TARDIS could work at various temperatures as long as changes in the temperature are compensated for. For the TARDIS, this compensation is done by using temperature sensors which are available in many of the today's microcontrollers.[5]

Impact of Additional Capacitance: Capacitors can greatly extend the resolution time of the TARDIS. In our experiment, we have tested five different capacitors ranging from 10 μF to 10 mF at 26.5°C. For this experiment, the capacitors were fully charged in the circuit and their voltage decay traces were recorded. These traces were later used in conjunction with our previous remanence-vs.-decay results (Section 6) to calculate the time frame

[5]According to the TI website, 37% of their microcontrollers are equipped with temperature sensors.

Cap. Size	Stage 1 (s)	Stage 2 (s)
0 μF	1.22e0	8.80e-1
10 μF	1.75e2	5.00e1
100 μF	1.13e3	8.47e2
1000 μF	1.17e4	9.50e3
10000 μF	1.43e5	>5.34e4*

* Test was interrupted.

Table 5: Estimated time in Stage 1 and Stage 2 of the TARDIS increases as capacitor size increases. The experiments are done on a MSP430F2131 microcontroller at 26.5°C and an SRAM size of 256 B. Stage 1 is the time after the power failure but before the SRAM decay. Stage 2 represents the duration of SRAM decay.

achievable with each capacitor. Table 5 summarizes the results for the duration of TARDIS Stage 1 and 2 based on capacitor size. The voltage decay traces, our conversion function (DRV Prob.), and the resulting SRAM-decay-over-time graph can be seen in Figure 10.

Results ranging from seconds to days open the path for a wide variety of applications for the TARDIS, as it can now be tweaked to work in a specific time frame. Current RFID-scale devices generally use capacitors ranging from tens of picofarads to tens of microfarads (e.g., [2] [3]). Although a 10 mF capacitor size might be large compared to the size of today's transiently powered devices, the progress in capacitors' size and capacity may very well make their use possible in the near future.

Impact of SRAM Size: Our hypothesis is that SRAM size has an inverse relation with decay time. This is ex-

Accepted to appear at the 21st USENIX Security Symposium, August 2012.

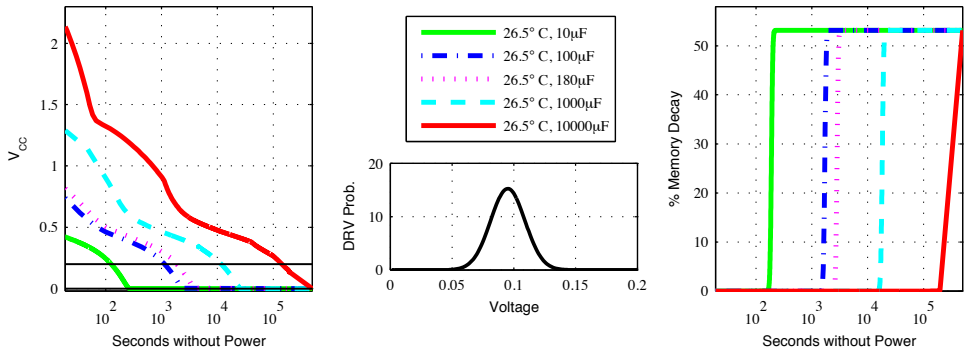

Figure 10: For five different capacitor values, measured supply voltage traces are combined with a pre-characterized DRV distribution to predict decay as a function of time. The decaying supply voltages after power is turned off are shown at left. The known DRV probabilities (Equation 4) for 26.5°C are shown at center. Equation 5 maps every supply voltage measurement to a predicted decay, thus creating the memory-decay-vs.-time plots shown at right. The two horizontal lines in the left image at approximately 150 and 50 mV are the voltages where the first and last bits of SRAM will respectively decay.

pected because a larger SRAM will have a larger leakage current and thus will drain the capacitor more quickly. We tested three different models of MSP430 microcontroller with SRAM sizes of 256 B, 2 KB, and 8 KB at 28°C with no capacitor. The DAQ sweeps off-time from 0 to a maximum of 5 seconds. The experiment results are consistent with our hypothesis and are shown in Figure 11. It should be noted that SRAM size is not the only difference between these three models, as they also have slightly different power consumptions.

Impact of Chip Variation: The chip-to-chip variation of the same microcontroller model is not expected to have a major effect on the TARDIS. We tested three instances of the MSP430F2131 with 256 B of memory and no capacitor at 27°C. The off-time changes from 0 to a maximum of 2.5 seconds with increments of 0.2 seconds. The result shown in Figure 12 matches our expectation and shows that changes in decay time due to chip-to-chip variation are insignificant (notice that no capacitor is used and the temperature for one of the chips is one degree higher). This result indicates that TARDIS would work consistently across different chips of the same platform and can be implemented on a system without concern for chip-to-chip variation.

TARDIS Simulation: We verified the TARDIS mechanism using SPICE simulation of a small SRAM array of 50 cells; the transistor models are 65 nm PTM, the power pin is connected to V_{CC} through a D1N4148 diode, and the decoupling capacitor is 70 nF. Each transistor is assigned a random threshold voltage deviation chosen uniformly from range ± 100 mV. Each line in Figure 13 plots the voltage difference across the two state nodes A and B for a single SRAM cell. Because all state nodes remain between 0V and V_{CC} during the discharge, the differential voltage is roughly enveloped by $\pm V_{CC}$ as shaded in grey. A positive differential voltage indicates a stored state of 1 (the written state), and a negative differential is a state of 0. Some of the nodes are observed to flip state, starting when V_{CC} reaches 200 mV at 0.55 seconds after power is disconnected. As V_{CC} discharges further, more cells decay by crossing from state 1 to 0. When V_{CC} is powered again at 1.05 seconds, each cell locks into its current state by fully charging either A or B and discharging the other; this is observed in Figure 13 as an increase in the magnitude of the differential voltage of each cell.

7 Inside an SRAM Cell

Each SRAM cell holds state using two cross-coupled inverters as shown in Figure 14; the access transistors that control reading and writing to the cell are omitted from the figure. The cross-coupled inverters are powered via connections to the chip's power supply node. The two states of the SRAM cell, representing a logical 1 and logical 0, are symmetrical. In each state, under normal conditions, the voltage of either A or B is approximately V_{cc} while the voltage of the other is approximately $0V$.

Data Retention Voltage: The minimum voltage at which each cell can store either a 0 or 1 is referred to as the cell's data retention voltage (DRV) [36]. Since DRV depends on random process variation, any set of SRAM cells will have a distribution of DRVs. Although the actual DRV distribution depends on process and design parameters, typical values fall within the range of 50 mV to 250 mV; a published design in 0.13 μm has a distribution of DRVs ranging from 80 mV to 250 mV, and our own analysis in

Accepted to appear at the 21st USENIX Security Symposium, August 2012.

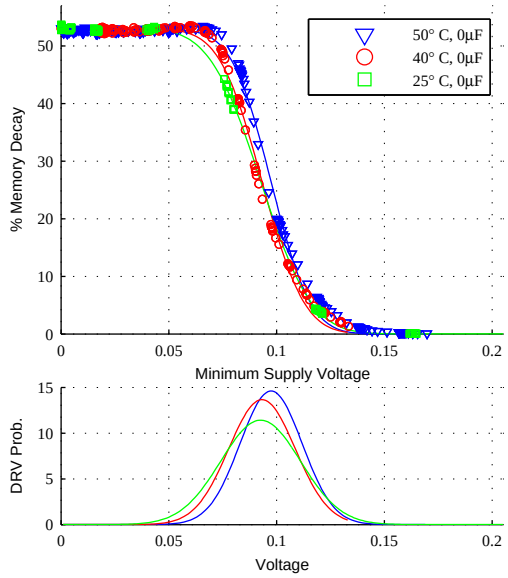

Figure 8: Regardless of temperature, the amount of decay depends almost entirely on the minimum supply voltage reached during a power-down. The bottom graph shows the 3-parameter DRV probabilities (Equation 4) that best predict the observed relationships between decay and minimum supply voltage for each of the three temperatures. The fit lines in the upper graph show the relationships between decay and minimum supply voltage that are predicted by these DRV models (Section 10).

this work estimates a majority of DRVs to be in the range of 50 mV to 160 mV (Figure 8).

7.1 Memory Decay Mechanisms

Memory decay occurs in SRAM when a cell loses its state during a power cycle and subsequently initializes to the opposite state upon restoration of power. Given that each cell typically favors one power-up state over the other [23, 17], memory decay can be observed only when the last-written state opposes the favored power-up state. We denote the favored power-up state as the *ground state*, since this is the value an SRAM cell will take at power-up after a very long time without power. We say that a cell written with the value opposite its ground state is *eligible* for memory decay. Each eligible cell will decay once the supply voltage falls below the cell's DRV. Cells that are randomly assigned very low DRVs thus do not decay until the supply voltage is very low. With sufficient capacitance, it can take days for all eligible cells to decay.

Supply voltage decays according to Equation 1, where V_{CC}, I_{CC}, and C_{CC} represent the supply voltage, current,

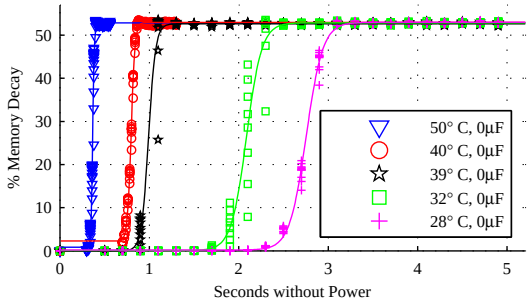

Figure 9: The duration of SRAM decay is non-zero across all temperatures even when no capacitor is used. For any given temperature, the duration of SRAM decay is consistent across trials. Increasing the temperature from 28°C to 50°C reduces the duration of both Stage 1 and Stage 2 decay by approximately 80%.

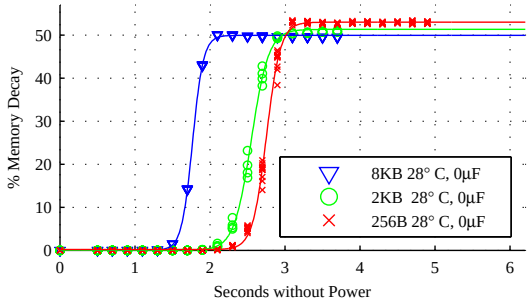

Figure 11: Different microcontrollers within the TI MSP430 family with different SRAM sizes exhibit different decay times, but follow the same general trend. The MSP430F2618, MSP430F169, and MSP430F2131 respectively have 8 KB, 2 KB, and 256 B of SRAM.

and capacitance of the power supply node. The voltage decay is slowed by a large capacitance and low current, and the following paragraphs explain why both are present in our TARDIS application.

$$\frac{dv_{CC}}{dt} = \frac{I_{CC}}{C_{CC}} \quad (1)$$

Large Capacitance: The large amount of charge stored on the power supply node is due to the decoupling capacitance that designers add between V_{CC} and gnd. During normal operation, this capacitance serves to stabilize the supply voltage to the functional blocks of the chip, including SRAM. In some experiments, the time ranges measurable by the TARDIS are further extended by supplementing the standard decoupling capacitors with additional explicit capacitance.

Accepted to appear at the 21st USENIX Security Symposium, August 2012.

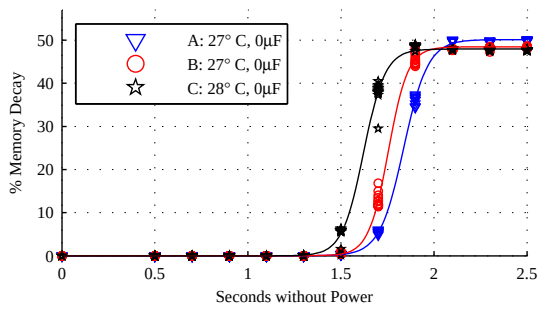

Figure 12: Decay versus time in 3 different instances of the MSP430F2131 microcontroller at similar temperatures. The durations of Stage 1 and Stage 2 decay match closely across instances.

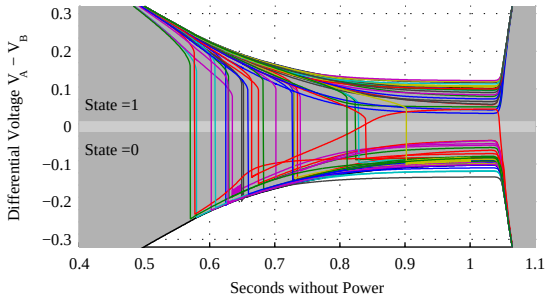

Figure 13: The differential voltage of SRAM cells during decay. The envelope of $\pm V_{CC}$ is shaded in grey. All cells are in the 1 state when power is first turned off. As V_{CC} decays, some cells flip from 1 to 0. The cells stabilize when power is restored. The number of zeros after the restoration of power is used to estimate the duration of the power outage.

Low Leakage Current: The total current I_{CC} comprises the operating current of the microcontroller and the SRAM's data-retention current; both currents are functions of the supply voltage. The current during the voltage decay is shown in Figure 15, and explained here:

Immediately after power is disconnected, supply voltages are above 1.4 V and the microcontroller is operational. The observed current is between 250 μA and 350 μA, consistent with the 250 μA current specified for the lowest-power operating point (1.8 V with 1 MHz clock) of the MSP430F2131 [50]. The SRAM current is negligible by comparison. The high current consumption causes the voltage to decay quickly while the microcontroller remains active.

As the voltage drops below 1.4 V, the microcontroller deactivates and kills all clocks to enter an ultra-low power RAM-retention mode in an attempt to avoid losing data.

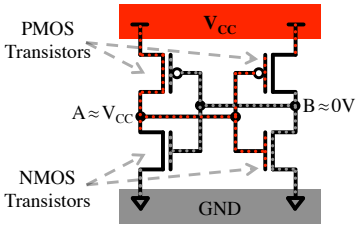

Figure 14: The state-holding portion of an SRAM cell consists of two cross-coupled inverters tied to the chip's power and ground nodes.

The nominal current consumed in this mode is only the data-retention current, specified to be 0.1 μA for the 256 B of SRAM in the MSP430F2131 [50]. In our observations, I_{CC} is between 0.5 μA and 10 μA during the time that V_{CC} is between 0.5 V and 1.4 V. This current is $1.5-3$ orders of magnitude smaller than the current when the microcontroller is active. With so little current being consumed, the supply voltage decays very slowly. The current further decreases as the supply voltage drops into subthreshold, and cells begin to experience memory decay.[6]

Impact of Temperature: Increasing the temperature leads to more rapid memory decay for two reasons. First, increasing the temperature increases the leakage currents that persist through data-retention mode. Increased leakage currents lead to a faster supply voltage decay, causing the supply voltage to drop below DRVs sooner. Second, temperature expedites memory decay by increasing the DRV of SRAM cells [36], causing them to decay at slightly higher supply voltages. Prior work shows a modest 13mV increase in DRV when temperature increases from 27°C to 100°C [36].

7.2 Choosing a State to Write

It is possible to increase the maximum observable memory decay by making every cell eligible for decay. This would be accomplished by characterizing the ground state of each SRAM cell over many remanence-free trials [17, 23], and then writing each cell with its non-ground state in order to make its memory decay observable. In contrast to writing a uniform 1 to all cells, this approach can extract more timing information from the same collection of SRAM cells. However, this alternative requires storing the ground states in non-volatile memory (or equivalently storing written states in non-volatile memory) in order to

[6]Note that setting V_{CC} to 0 V during the power-down, instead of leaving it floating, reduces voltage and memory decay times by at least an order of magnitude [44] by providing a low impedance leakage path to rapidly drain the capacitance; we have observed this same result in our experiments as well.

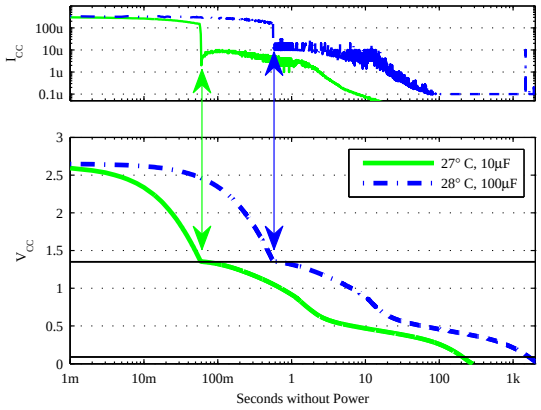

Figure 15: Supply voltage and current during two power-down events with different capacitors. The voltage V_{CC} is measured directly, and the current I_{CC} is calculated per Equation 1 using the measured $\frac{dV_{CC}}{dt}$ and known capacitor values. The voltage initially decays rapidly due to the high current draw of the microcontroller. When V_{CC} reaches $1.40V$ the microcontroller turns off and I_{CC} drops by several orders of magnitude, leading to a long and slow voltage decay. At the time when V_{CC} crosses the horizontal line at 0.09V, approximately half of all eligible cells will have decayed.

evaluate whether or not a cell has decayed. Our approach of writing a uniform 1 to all cells makes it possible to evaluate memory decay without this overhead simply by evaluating the Hamming Weight of the SRAM state.

8 Alternative Approaches

The more general question of how to keep time without a power source is fundamental and has numerous applications in security and real-time computing. Techniques for keeping time without power or with very reduced power typically rely on physical processes with very long time constants. In CMOS, the most obvious process with a long time constant is the leakage of charge off of a large capacitor through a reverse-biased diode or MOSFET in the cut-off region.

An unexplored alternative to the TARDIS is charging a capacitor whenever the device is active, and checking the capacitor's voltage at a subsequent power-up to determine whether the device has been active recently. The power-up measurement can be performed using an ADC if available, or else by checking whether or not the remaining voltage is sufficient to register as a logical 1. This approach differs from the TARDIS in incurring monetary and power costs due to the use of a dedicated capacitor and dedicated input-output pins for charging the capacitor and sensing its voltage. Furthermore, the capacitor voltage is still dynamic after power-up, leaving the measurement sensitive to timing variations caused by interrupts. By comparison, the TARDIS uses no dedicated capacitor or input-output pins; its measurement materializes in SRAM at power-up and remains static thereafter until being read and subsequently overwritten.

The EPC Gen2 protocol [12] requires UHF RFID tags to maintain four floating-gate based "inventorial flags" used to support short power gaps without losing the selected/inventoried status. An interesting alternative approach could co-opt these flags to provide a notion of time; however, the flags only persist between 500ms and 5s across power failures. In comparison, the SRAM-based approach in the TARDIS has a resolution time from seconds to hours and has a temperature compensation mechanism. Another advantage of the TARDIS is that it works on any SRAM-based device regardless of the existence of special circuits to support inventorial flags.

9 Related Work

RFID Security and Privacy: The inability of intermittently powered devices to control their response rates has made them susceptible to various attacks. An RFID tag could be easily "killed" by exhausting all possible 32-bit "kill" keys. Such unsafe "kill" commands could be replaced with a "sleep" command [25]; however, lack of a timer to wake up the tag in time has made the use of the "sleep" command inconvenient. The key to e-passports can be discovered in real time by brute-force attacks [6]. The attack could be slowed down if the e-passport had a trustworthy notion of time. The minimalist model [24] offered for RFID tags assumes a scheme that enforces a low query-response rate. This model could be implemented using the TARDIS.

Secure Timers: To acquire a trustworthy notion of time, multiple sources of time can be used to increase the security level of a timer [40]; but this requires the device to interact actively with more than one source of time, which is not practical for RFID tags that use passive radio communication. The same issues prevent us from using the Lamport clock and other similar mechanisms that provide order in distributed systems [26]. This inability to acquire secure time precludes the use of many cryptographic protocols, including timed-release cryptography [29] [39]

Ultra-low Power Clocks: With the rise of pervasive computing come a need for low-power clocks and counters. Two example applications for low-power clocks are timestamping secure transactions and controlling when a device

Accepted to appear at the 21st USENIX Security Symposium, August 2012.

should wake from a sleep state. The lack of a rechargeable power source in some pervasive platforms requires ultra-low power consumption. Low voltage and subthreshold designs have been used to minimize power consumption of digital circuits since the 1970s [48]. Circuits in wristwatches combine analog components and small digital designs to operate at hundreds of nW [53]. A counter designed for smart cards uses adiabatic logic to operate at 14KHz while consuming 11nW of power [49]. A gate-leakage-based oscillator implements a temperature-invariant clock that operates at sub-Hz frequencies while consuming 1pW at 300mV [28]. A TI-recommended technique [37] for the MSP430 is to charge a dedicated external capacitor from the microcontroller while in a low-power sleep mode with clocks deactivated; the microcontroller is triggered to wake up when the capacitor voltage surpasses a threshold. But all of these solutions, while very low-power, still require a constant supply voltage and hence a power source in the form of a battery or a persistently charged storage capacitor. However, embedded systems without reliable power and exotic low-power timers may still benefit from the ability to estimate time elapsed since power-down.

Attacks Based on Memory Remanence: Processes with long time constants can also raise security concerns by allowing data to be read from supposedly erased memory cells. Drowsy caches [13] provide a good background on the electrical aspects of data retention. Gutmann stated that older SRAM cells can retain stored state for days without power [18]. Gutmann also suggest exposing the device to higher temperatures to decrease the retention time. Anderson and Kuhn first proposed attacks based on low-temperature SRAM data remanence [4]. Experimental data demonstrating low-temperature data remanence on a variety of SRAMs is provided by Skorobogatov [44], who also shows that remanence is increased when the supply during power-down is left floating instead of grounded. More recent freezing attacks have been demonstrated on a 90nm technology SRAM [52], as well as on DRAM [19]. Data remanence also imposes a fundamental limit on the throughput of true random numbers that can be generated using power-up SRAM state as an entropy source [42]. The TARDIS, in finding a constructive use for remanence and decay, can thus be seen as a counterpoint to the attacks discussed in this section. The TARDIS is the first *constructive* method that takes advantage of SRAM remanence to increase the security and privacy of intermittently powered devices.

10 Conclusions

A trustworthy source of time on batteryless devices could equip cryptographic protocols for more deliberate defense against semi-invasive attacks such as differential power analysis and brute-force attacks. The TARDIS uses remanence decay in SRAM to compute the time elapsed during a power outage—ranging from seconds to hours depending on hardware parameters. The mechanism provides a coarse-grained notion of time for intermittently powered computers that otherwise have no effective way of measuring time. Applications using the TARDIS primarily rely on timers with hourglass-like precision to throttle queries. The TARDIS consists purely of software, making the mechanism easy to deploy on devices with SRAM. A novel aspect of the TARDIS is its use of memory decay or data remanence for improved security rather than attacking security. Without the TARDIS, batteryless devices are unlikely to give you the time of day.

Acknowledgments

The authors would like to thank our shepherd Jonathan McCune; Gesine Hinterwalter, Karsten Nohl, David Oswald, and Joshua Smith for their feedback on applications; Gildas Avoine for information on passport communication and feedback on applications; Matt Reynolds for information on the EPC gen2 protocol; Quinn Stewart for proofreading; and members of the UMass SPQR lab for reviewing early versions of this paper.

This research is supported by NSF grants CNS-0831244, CNS-0845874, CNS-0923313, CNS-0964641, SRC task 1836.074, Gigascale Systems Research Center, and a Sloan Research Fellowship. Any opinions, findings, conclusions, and recommendations expressed in these materials are those of the authors and do not necessarily reflect the views of the sponsors. Portions of this work are patent pending.

References

[1] The TARDIS, British Broadcasting Channel. `http://www.bbc.co.uk/doctorwho/characters/tardis.shtml`, November 1963.

[2] Hpc0402b/c - high performance, high precision wire-bondable 0402 capacitor for smartcard, high-frequency and substrate-embedded applications. `http://www.vishay.com/docs/10120/hpc0402b.pdf`, Dec. 2008.

[3] An introduction to the architecture of Moo 1.0. `https://spqr.cs.umass.edu/moo/Documents/Moo_01242011.pdf`, May 2011.

[4] ANDERSON, R., AND KUHN, M. Tamper resistance: a cautionary note. In *Proceedings of the 2nd USENIX Workshop on Electronic Commerce* (1996).

[5] AVOINE, G. Personal communication on French passports. 2012.

[6] AVOINE, G., KALACH, K., AND QUISQUATER, J.-J. ePassport: Securing international contacts with contactless chips. In *Financial Cryptography and Data Security* (2008), G. Tsudik, Ed., Springer-Verlag, pp. 141–155.

[7] BONO, S., February 2012. Personal communication.

Accepted to appear at the 21st USENIX Security Symposium, August 2012.

[8] BONO, S. C., GREEN, M., STUBBLEFIELD, A., JUELS, A., RUBIN, A. D., AND SZYDLO, M. Security analysis of a cryptographically-enabled RFID device. In *Proceedings of the 14th USENIX Security Symposium* (2005).

[9] BUETTNER, M., GREENSTEIN, B., WETHERALL, D., AND SMITH, J. R. Revisiting smart dust with RFID sensor networks, 2008.

[10] CANTHERM. Thermal cut-offs. `http://www.cantherm.com/products/thermal_fuses/sdf.html`, 2011. Last Viewed May 14, 2012.

[11] CHOTHIA, T., AND SMIRNOV, V. A traceability attack against e-Passports. In *14th International Conference on Financial Cryptography and Data Security* (2010), Springer.

[12] EPCGLOBAL. *EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communication at 860 MHZ–960 MHZ, Version 1.2.0*.

[13] FLAUTNER, K., KIM, N. S., MARTIN, S., BLAAUW, D., AND MUDGE, T. Drowsy caches: simple techniques for reducing leakage power. In *Proc. 29th IEEE/ACM International Symposium on Computer Architecture* (2002), pp. 148–157.

[14] GANERIWAL, S., ČAPKUN, S., HAN, C.-C., AND SRIVASTAVA, M. B. Secure time synchronization service for sensor networks. In *Proceedings of the 4th ACM Workshop on Wireless Security* (2005), WiSe '05, pp. 97–106.

[15] GARCIA, F. D., ROSSUM, P. V., VERDULT, R., AND SCHREUR, R. Wirelessly pickpocketing a MIFARE Classic card. In *IEEE Symposium on Security and Privacy* (May 2009), pp. 3–15.

[16] GOLDBERG, I., AND BRICENCO, M. GSM cloning. `http://www.isaac.cs.berkeley.edu/isaac/gsm-faq.html`, 1999. Last Viewed February 19, 2012.

[17] GUAJARDO, J., KUMAR, S., SCHRIJEN, G., AND TUYLS, P. FPGA intrinsic PUFs and their use for IP protection. In *Cryptographic Hardware and Embedded Systems (CHES)* (2007), pp. 86–80.

[18] GUTMANN, P. Secure deletion of data from magnetic and solid-state memory. In *Proceedings of the 6th USENIX Security Symposium* (Jan 1996).

[19] HALDERMAN, J., SCHOEN, S., HENINGER, N., CLARKSON, W., PAUL, W., CALANDRINO, J., FELDMAN, A., APPELBAUM, J., AND FELTEN, E. Lest we remember: Cold boot attacks on encryption keys. In *Proceedings of the 17th USENIX Security Symposium* (2008).

[20] HALPERIN, D., HEYDT-BENJAMIN, T. S., RANSFORD, B., CLARK, S. S., DEFEND, B., MORGAN, W., FU, K., KOHNO, T., AND MAISEL, W. H. Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In *Proceedings of the 29th Annual IEEE Symposium on Security and Privacy* (May 2008), pp. 129–142.

[21] HEYDT-BENJAMIN, T. S., BAILEY, D. V., FU, K., JUELS, A., AND OHARE, T. Vulnerabilities in first-generation RFID-enabled credit cards. In *Proceedings of Eleventh International Conference on Financial Cryptography and Data Security, Lecture Notes in Computer Science, Vol. 4886* (February 2007), pp. 2–14.

[22] HO, H., SAEEDI, E., KIM, S., SHEN, T., AND PARVIZ, B. Contact lens with integrated inorganic semiconductor devices. In *Micro Electro Mechanical Systems, 2008. MEMS 2008. IEEE 21st International Conference on* (Jan. 2008), pp. 403–406.

[23] HOLCOMB, D. E., BURLESON, W. P., AND FU, K. Power-up SRAM state as an identifying fingerprint and source of true random numbers. *IEEE Transactions on Computers* (2009).

[24] JUELS, A. Minimalist cryptography for low-cost RFID tags (extended abstract). In *Security in Communication Networks*, C. Blundo and S. Cimato, Eds., vol. 3352 of *Lecture Notes in Computer Science*. Springer, 2005, pp. 149–164.

[25] JUELS, A. RFID security and privacy: A research survey. *IEEE Journal on Selected Areas in Communications 24*, 2 (February 2006), 381–394.

[26] LAMPORT, L. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM 21*, 7 (1978), 558–565.

[27] LEWIS, P. H. Of privacy and security: The clipper chip debate. The New York Times, April 24, 1994.

[28] LIN, Y., SYLVESTER, D. M., AND BLAAUW, D. T. A sub-pW timer using gate leakage for ultra low-power sub-Hz monitoring systems. *Custom Integrated Circuits Conference* (2007).

[29] MAO, W. Timed-release cryptography. In *Selected Areas in Cryptography VIII (SAC'01* (2001), Prentice Hall, pp. 342–357.

[30] MCGRAW, G. Silver bullet podcast: Interview with Ross Anderson. `http://www.cigital.com/silver-bullet/show-070/`. Show #70, January 31, 2012.

[31] NXP Semiconductors MIFARE classic. `http://www.nxp.com/products/identification_and_security/smart_card_ics/mifare_smart_card_ics/mifare_classic/`. Last Viewed February 18, 2012.

[32] NXP Semiconductors SPI real time clock/calendar. `http://www.nxp.com/documents/data_sheet/PCF2123.pdf`. Last Viewed February 18, 2012.

[33] OMEGA ENGINEERING, I. *OSXL450 Infrared Non-Contact Thermometer Manual*.

[34] OREN, Y., AND SHAMIR, A. Remote password extraction from RFID tags. *Computers, IEEE Transactions on 56*, 9 (Sept. 2007), 1292–1296.

[35] OSWALD, D., AND PAAR, C. Breaking MIFARE DESFire MF3ICD40: Power analysis and templates in the real world. In *Cryptographic Hardware and Embedded Systems (CHES)* (2011), pp. 207–222.

[36] QIN, H., CAO, Y., MARKOVIC, D., VLADIMIRESCU, A., AND RABAEY, J. SRAM leakage suppression by minimizing standby supply voltage. In *Proceedings of 5th International Symposium on Quality Electronic Design* (2004), pp. 55–60.

[37] RAJU, M. UltraLow Power RC Timer Implementation using MSP430. In *Texas Instruments Application Report SLAA119* (2000).

[38] RANSFORD, B., CLARK, S., SALAJEGHEH, M., AND FU, K. Getting things done on computational RFIDs with energy-aware checkpointing and voltage-aware scheduling. In *USENIX Workshop on Power Aware Computing and Systems (HotPower '08)* (Dec. 2008).

[39] RIVEST, R. L., SHAMIR, A., AND WAGNER, D. A. Time-lock puzzles and timed-release crypto. Tech. rep., Cambridge, MA, USA, 1996.

[40] ROUSSEAU, L. Secure time in a portable device. In *Gemplus Developer Conference* (2001).

[41] SAMPLE, A. P., YEAGER, D. J., POWLEDGE, P. S., MAMISHEV, A. V., AND SMITH, J. R. Design of an RFID-based battery-free programmable sensing platform. *IEEE Transactions on Instrumentation and Measurement 57*, 11 (Nov. 2008), 2608–2615.

[42] SAXENA, N., AND VORIS, J. We can remember it for you wholesale: Implications of data remanence on the use of RAM for true random number generation on RFID tags. In *Proceedings of the Conference on RFID Security* (2009).

Accepted to appear at the 21st USENIX Security Symposium, August 2012.

[43] SCHNEIER, B. *Applied cryptography (2nd ed.): Protocols, algorithms, and source code in C*. John Wiley & Sons, Inc., 1995.

[44] SKOROBOGATOV, S. Low temperature data remanence in static RAM. Tech. Rep. UCAM-CL-TR-536, University of Cambridge Computer Laboratory, 2002.

[45] STAJANO, F., AND ANDERSON, R. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Security Protocols*, B. Christianson, B. Crispo, J. Malcolm, and M. Roe, Eds., vol. 1796 of *Lecture Notes in Computer Science*. Springer, 2000, pp. 172–182.

[46] SUN, K., NING, P., AND WANG, C. TinySeRSync: secure and resilient time synchronization in wireless sensor networks. In *Proceedings of the 13th ACM Conference on Computer and Communications Security* (2006), CCS '06, pp. 264–277.

[47] SUN ELECTRONIC SYSTEMS, I. *Model EC1X Environmental Chamber User and Repair Manual*, 2011.

[48] SWANSON, R., AND MEINDL, J. D. Ion-implanted complementary MOS transistors in low-voltage circuits. *International Solid-State Circuits Conference* (May 1972).

[49] TESSIER, R., JASINSKI, D., MAHESHWARI, A., NATARAJAN, A., XU, W., AND BURLESON, W. An energy-aware active smart card. *IEEE Transaction on Very Large Scale Integration (VLSI) Systems* (2005).

[50] TEXAS INSTRUMENTS INC. MSP430F21x1 Mixed Signal Microcontroller. In *Texas Instruments Application Report SLAS439F* (Sep. 2004, revised Aug. 2011).

[51] THINGMAGIC INC. *Mercury 4/ MERCURY 5 User Guide*, February 2007.

[52] TUAN, T., STRADER, T., AND TRIMBERGER, S. Analysis of data remanence in a 90nm FPGA. *Custom Integrated Circuits Conference* (2007).

[53] VITTOZ, E. Low-power design: Ways to approach the limits. *International Solid-State Circuits Conference* (May 1994).

[54] XU, X., GU, L., WANG, J., AND XING, G. Negotiate power and performance in the reality of RFID systems. In *PerCom* (2010), IEEE Computer Society, pp. 88–97.

[55] YEAGER, D., ZHANG, F., ZARRASVAND, A., GEORGE, N., DANIEL, T., AND OTIS, B. A 9 μa, addressable Gen2 sensor tag for biosignal acquisition. *IEEE Journal of Solid-State Circuits 45*, 10 (Oct. 2010), 2198–2209.

[56] ZHANG, H., GUMMESON, J., RANSFORD, B., AND FU, K. Moo: A batteryless computational RFID and sensing platform. Tech. Rep. UM-CS-2011-020, Department of Computer Science, University of Massachusetts Amherst, Amherst, MA, June 2011.

Appendix

Model of Decay Probabilities

Knowing the DRV distribution of a collection of SRAM cells makes it possible to predict the amount of memory decay that will result from reaching any known minimum supply voltage during a power cycle. We propose a simple and intuitive 3-parameter (α, μ, σ) model to characterize the DRV distribution. We chose the parameters such that the model predictions agree with empirical data relating memory decay to minimum supply voltage.

Cells eligible for memory decay after being written with a value of 1 are those with a ground state of 0. We use $g = 0$ to denote cells with a 0 ground state, and use α to denote the fraction of cells with this ground state; α is therefore the largest fraction of cells that can decay after writing a 1 to all cells.

$$\Pr(g = 0) = \alpha \tag{2}$$

Among cells that are eligible for memory decay, we assume that DRVs are normally distributed with mean μ and standard deviation σ (Equation 3).

$$DRV \mid (g = 0) \sim \mathcal{N}\left(\mu, \sigma^2\right) \tag{3}$$

The probability of a randomly selected cell being eligible for memory decay and having $DRV = v$ is given by Equation 4. This is an α-scaled instance of the PDF of a normally distributed random variable, and we refer to this as the "DRV probability" of voltage v.

$$\Pr\left((g = 0) \wedge (DRV = v)\right) = \frac{\alpha}{\sigma\sqrt{2\pi}} e^{-(v-\mu)^2/\left(2\sigma^2\right)} \tag{4}$$

If the minimum voltage of a power cycle is known, then the 3-parameter model can predict the memory decay. The cells that will decay are eligible cells with a DRV that is above the minimum supply voltage reached during the power cycle. A closed-form equation for predicting the memory decay from the minimum voltage and model parameters is then given by Equation 5; this equation is 1 minus the CDF of a normally distributed random variable, scaled by α.

$$D_{PRED}(v_{min}, \alpha, \mu, \sigma) = \alpha\left(1 - \frac{1 + erf\left(\frac{v_{min}-\mu}{\sigma\sqrt{2}}\right)}{2}\right) \tag{5}$$

A 3-parameter model is evaluated according to how well its predicted memory decay matches empirical data. The evaluation is performed using a set of n observations $\langle v_0, D(v_0)\rangle, \langle v_1, D(v_1)\rangle, \ldots, \langle v_{n-1}, D(v_{n-1})\rangle$; each observation is a measurement of the minimum supply voltage reached during a power cycle, and the memory decay observed across that power cycle. The prediction error of any model is defined according to Equation 6. We initially use the set of measurements to find the model parameters that minimize the prediction error (see Figure 8).

$$ERR(\alpha, \mu, \sigma) = \sum_{i=0}^{n-1} \left(D_{PRED}\left(v_i, \alpha, \mu, \sigma\right) - D\left(v_i\right)\right)^2 \tag{6}$$

After measurements are used to fit the model parameters to empirical data, the model is subsequently used to predict memory-decay-vs.-time curves from voltage-vs.-time measurements (see Figure 10).

Accepted to appear at the 21st USENIX Security Symposium, August 2012.