Zhang, Ruide; Wang, Ning; Zhang, Ning; Yan, Zheng; Lou, Wenjing; Hou, Y. Thomas

PriRoster: Privacy-preserving Radio Context Attestation in Cognitive Radio Network

Published in:
IEEE International Symposium on Dynamic Spectrum Access Networks

DOI:
10.1109/DySPAN.2019.8935740

Published: 01/01/2019

Document Version
Peer reviewed version

PriRoster: Privacy-preserving Radio Context Attestation in Cognitive Radio Network

Ruide Zhang†, Ning Wang†, Ning Zhang*, Zheng Yan‡, Wenjing Lou†, and Y. Thomas Hou†

†Virginia Polytechnic Institute and State University, VA, USA
†{rdzhang,ning18,wjlou,thou}@vt.edu
*Washington University, St. Louis, MO, USA
*zhang.ning@wustl.edu
‡Xidian University, China & Aalto University, Finland
‡zyan@xidian.edu.cn

***Abstract*—Spectrum shortage is becoming a global concern and cognitive radio network (CRN) is envisioned to be one of the key technologies for overcoming this challenge. However, proper operation of CRN heavily depends on compliance of cognitive radios (CRs). Although Remote attestation of CRs' radio context is a promising solution, delegating appraisal tasks to local base stations has serious privacy concerns. Conducting appraisal tasks only on global appraiser brings an easy solution, nevertheless, scalability remains an unsolved issue.**

In this paper, we propose PriRoster, a privacy-preserving radio context attestation framework for cognitive radio network. The proposed framework takes advantage of recent advancement in trusted hardware, Intel SGX, and incorporate it in a secure and scalable way. First, we propose a privacy-preserving design for single device remote attestation. Second, we design a secure trust transfer scheme to delegate power consuming process of trust establishment between local appraiser (LA) enclave and CR nodes to global appraiser (GA). Through this design, we construct a scalable framework with low computation burden for resource constraint low-end CR devices. Furthermore, special considerations are given in adopting Intel SGX. We consider known memory access side channel of Intel SGX and propose oblivious appraisal functions to prevent this kind of information exposure. At last, we build a prototype of the proposed system using Raspberry Pi, USRP, Intel NUC and AWS cloud. The feasibility of our proposed framework is measured by system benchmark and the effectiveness of proposed oblivious appraisal functions are verified by dynamic code instrumentation.

I. Introduction

With the coming of tremendous amount of smart devices, the world has witnessed an increasing popularity of wireless services in the last decade. Wireless communities throughout the world have recognized the shortage of spectrum for commercial broadband uses and acknowledged the urgent need for an effort to make additional spectrum available for broadband data. Nevertheless, current fixed spectrum allocation methodology is wasting a great portion of spectrum resources. In fact, recent measurements by Federal Communications Commission (FCC) have shown that 70% of the allocated spectrum in US is not utilized [1].

In order to improve spectrum utilization rate in the generation wireless communication, the concept of CRN is proposed [2]. CRN provides opportunistic access to unused spectrum which allows secondary users (SUs) to utilize the radio spectrum allocated to the primary users (PUs) when the spectrum is temporally not being utilized. Wireless communities in the United States embrace the concept of CRN and propose tiered-access to shared spectrum. The Federal Communications Commission (FCC) has proposed a dynamic spectrum management framework for a Citizen Broadband Radio Service (CBRS) governed by a spectrum access system (SAS) [3]. The SAS would take inputs from PUs regarding their spectrum utilization and from radio environment map collected by sensing partners such as Google. Then, SAS manage the use of the available spectrum opportunities for SUs by granting transmission permits to a CR based on the its access level and location.

Nevertheless, proper operation of SAS relies heavily on honest CRs. Honest CRs comply to transmission permits they receive from SAS and report honestly to SAS when queried. To fulfill this requirement, [4] proposes remote attestation of CR according to its radio context. Remote attestation is the activity of making a claim about properties of a target by supplying evidence to an appraiser over a network [5]. Remote attestation of radio context provides reliable evidence about the state of software executing and about the wireless transmission context on a CR device. This evidence ensure that attested CR devices will not engage in some class of misbehavior.

Although remote attestation of radio context at CR seems promising, performing remote appraisals only at GA is not scalable. Thus, delegating appraisal tasks to edge base stations (BSs) is recommended. However, service providers who control the edge BSs have a notorious history in terms of privacy. For example, service providers (like AT&T, Sprint, Verizon, and T-Mobile) are known to collect data such as incoming and outgoing calls, locations [6]. They tracks "the webpages you visit, the time you spend on each, the links or ads you see and follow, and the search terms you enter" [7]. Service providers are also forced to provide data for governmental surveillance. For instance, the National Security Agency (NSA) has been secretly collecting the phone call records of tens of millions of Americans, using data provided by AT&T, Verizon and

BellSouth [8]. Therefore, adopting edge BS as LA brings serious privacy concerns. For example, leakage of location trajectory and transmission parameters is a serious concern for PUs that are sensitive federal and military devices [9]. And leakage of software configurations concerns both PUs and SUs, for knowing software configurations brings great advantages for malicious actors to find vulnerabilities in a CR device [10].

In this work, we present PRIvacy-preserving Radio cOntext atteSTation in cognitivE Radio network (PriRoster). To the best of our knowledge, this is the first work to provide scalable privacy-preserving remote attestation of radio context function for cognitive radio networks. We achieve the goal of preserving privacy of LAs on edge BSs by introducing trusted hardware, Intel Software Guard Extensions (SGX). Intel's SGX is a set of extensions to the Intel architecture that aims to solve the secure remote computation problem by leveraging trusted hardware in the remote computer [11]. While secure system built on top of Intel SGX is a relatively mature field, the integration of Intel SGX to preserve privacy in CRN radio context attestation is challenged by scalability requirement and by side channels on Intel SGX.

The first challenge is scalability when integrating Intel SGX. To make a CR device trust edge BS before it upload its attestation report, the CR device needs to perform a remote attestation on the SGX enclave inside edge BS. However, CR devices are resource constraint and periodically performing remote attestation on SGX enclave consumes non-negligible amount of energy. Furthermore, creating independent SGX enclaves for large amount of CR devices brings great computation burden on edge BSs. In PriRoster, the power consuming attestation on SGX enclaves is delegated to global appraiser and only one enclave is needed on each edge BS for conducting local appraisals.

The second challenge is the privacy leakage from memory access side channel on Intel SGX. Memory access side channel is a known vulnerability on Intel SGX [12]–[14]. A privileged software can observe the memory access pattern of an enclave executing. Through memory access side channel, useful information like outline of a picture can be inferred. In our case, edge BS can potentially use memory access pattern observed to infer the radio context of CR devices. In PriRoster, we design oblivious appraisal functions for preventing memory access pattern leakage.

To summarize, our contributions are:

- We design PriRoster, the first proposal for privacy-preserving radio contet attestation in cognitive radio network.
- We build a prototype of PriRoster using USRP, Raspberry Pi, Intel NUC and Amazon AWS. The prototype of the system demonstrates the feasibility of adopting PriRoster framework in large scale.
- We design oblivious appraisal functions for hiding memory access pattern from edge BS. We use Intel Pin tool to show the effectiveness of our proposed design.

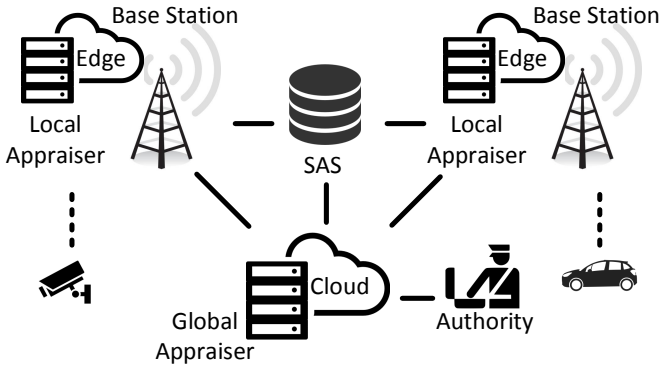

Fig. 1: Radio Context Attestation in CRN.

II. Background

A. Radio Context Attestation in CRN

Fig. 1 shows the architecture for facilitating radio context attestation in CRN. The CRN under consideration adopts centralized spectrum management as described in the CBRS whitepaper by the FCC [3]. In [4], the attestation procedures starts with an attestation request from authority to global appraiser. Then depends on distinct security level of SAS, the verification of radio context except software context can be delegated to LA or not. If the security level of SAS is low, the GA then sends the spectrum availability information to the BSs. After that, the BSs ask CR devices to generate attestation report according to the status of their radio context. Radio context includes software configuration, radio configuration, location and time. Upon receiving the attestation reports, BSs start local appraisal process to audit the reports and forward results back to GA. If the security level of SAS is high, spectrum availability information is not sent to BSs. And attestation report generated by CR devices are not audited at BSs. Instead, attestation reports are directly forwarded to GA for appraisal process.

However, in our case, PriRoster integrates trusted hardware to mitigate information leakage from edge BS, so we do not need two distinct design for different security level of SAS. Since under our framework, LAs are trustworthy, we can delegate appraisal tasks to edge BSs for either security level of SAS.

B. Intel SGX

Intel SGX is the latest Intel's instruction extensions that allows processes to shield part of their address space from privileged software such as operating system and hypervisor. Processes on SGX-capable platform can construct trusted execution environments called enclaves. Integrity and confidentiality guarantees are provided to security-sensitive computation conducted inside enclaves. Intel SGX also provides remote attest and provision, which allows a remote party like SAS or GA to verify an application enclave's identity and securely provision keys, credentials, and other sensitive data to enclave on edge BS.

Despite the security features brought by Intel SGX, there exists some known security limitations in modern Intel processors. Although MEE encrypts data in DRAM, if an attacker sniffs the address bus physically, he or she can observe a cache line-granularity side channel, which has been confirmed at both page [13] and cache line level [12].

III. System Model and Assumptions

Security Goals: PriRoster is designed to provide an authority the ability execute a network wide radio compliance measurement. In the meantime, PriRoster targets at preventing edge BS from obtaining knowledge about CR execution states and compliance rules. This goal is achieved by using two distinct remote attestation for CR device and edge BS respectively as the building block. Aggregated attestation report on all the nodes in CRN ensures authority the radio context of the radio nodes as well as their compliance to the spatial-temporal sensitive radio policy. Remote attestation of LA enclave mitigates privacy concerns on edge BS.

Threat Model: For the CR devices, we assume attacker can conduct software attacks to gain control of the CR device, therefore can modify the radio related parameters like the transmission power, the modulation method and more. He or she can also fabricate any network packets coming out of the controlled device. We do not consider hardware attacks. For the edge BSs, we assume there could be an malicious actor like a malicious insider or a remote attacker controlling its computing platform. The malicious actor can reveal any information sent to edge BS and fabricate any network packets sent out of the edge BS.

Assumptions: We assume CRs are equipped with trusted hardware components like widely available ARM TrustZone [15]. We assume CRs' software stack contains normal world and secure world. And the integrity of secure world software is guaranteed by secure boot. We certificate of GA is predistributed in secure world of CR nodes. And certificates of CR nodes are predistributed in GA. We assume remote attestation report generation is sitting inside trusted hardware and software attack cannot reveal or modify the process. We assume CR devices are powerful enough to perform asymmetric cryptograhic primitives. For the edge BSs, we assume they are equipped with Intel SGX [11]. We assume edge BSs can only control the privileged software like hypervisor and operating system but cannot modify hardware. We assume edge BSs can use privileged software to observe fine-grained memory trace.

IV. PriRoster Framework

Our goal is to design a network-wide radio context attestation framework that allows secure and scalable verification of operational integrity for a large number of CR devices in a spectrum sharing network. In order to keep the framework scalable, radio context appraisal of CR nodes has to be done at edge BSs while only aggregated attestation results are sent back to GA. Since radio context of CR node contains sensitive information, local appraisal at BS should not reveal actual radio context on CR nodes to edge BS. Besides, SAS compliance rules used in local appraisal needs protection since this information can be used to infer sensitive information like location of military radio. Thus, in our target framework, compliance check should not reveal compliance rules to edge BS. There exists three major challenges in achieving our goal:

1) Conducting local appraisal at untrustworthy edge nodes leaks sensitive information including radio context and compliance rules. To provide privacy-preserving radio context attestation for CR nodes, a straightforward design by integrating Intel SGX is demonstrated in **Sec. IV-A**.
2) Setting up a single enclave on edge BS for every CR node can preserve privacy, however, it cannot scale up. One potential solution is to implement one enclave on each BS for all devices connected to this BS. However, this design still requires performing remote attestation of the LA enclave on CR device for each attestation request from authority. This lead to a non-negligible energy consumption at the CRs side and tremendous amount of attestation burden on Intel Attestation Service (IAS) server. We propose a trust transfer design for CR node to delegate remote attestation to GA in order to solve this challenge as given in **Sec. IV-B**.
3) Intel SGX provides confidentiality and integrity for enclave programs, however, there exists some known security limitations on it. Although privileged software cannot access enclave memory, it can be used to observe memory access pattern. Therefore, an attacker controlling privileged software can potentially discloses sensitive information such as software configuration of CR. To mitigate this kind of side channel attack, we realize oblivious software configuration appraisal by designing oblivious function in **Sec. IV-C**.

A. Privacy-Preserving Device Attestation

In this section, we present a straightforward design for privacy preserving remote attestation of radio context on a single device as shown in Fig. 2 following step ① to ⑨. In comparison with previous work [4], we consider edge BS not trusted. We take advantage of trusted hardware (i.e. Intel SGX enclave) for defending against malicious edge BS. From high level view, when receiving a radio context attestation request, CR node remote attest LA enclave created by BS with the help of IAS server. If this attestation succeeds, integrity and confidentiality of LA enclave is protected by Intel SGX. Then CR sends its radio context attestation report to LA enclave for local appraisal. Note that, SAS and GA needs to perform remote attestation on LA enclave also and send radio context attestation material to LA enclave if IAS reports the LA enclave is trustworthy. We skip this part for simplicity of demonstration. In the end, LA enclave sends local appraisal result to GA. We show detail procedures of this scheme next.

As shown in Fig. 2, when authority initiates an radio context attestation, GA first pushes remote attestation request to BSs in step ①. Then BSs forward the request to CRs in step ②. Upon receiving the request, a CR in turn requests to

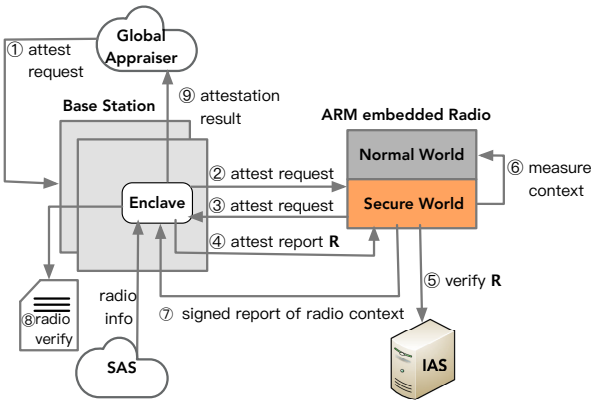

Fig. 2: Privacy-Preserving Device Attestation

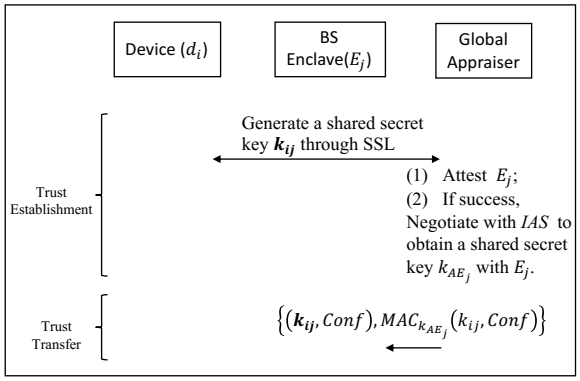

Fig. 3: Trust establishment and trust transfer procedure.

attest the SGX enclave on the BS to which it connects in step ③. SGX enclave on the BS sends back its enclave attestation report R to CR node in step ④. With the help of IAS, CR verifies the report received in step ⑤. Only if a success verification from IAS is received, will CR do radio context measurement in step ⑥. Then the attestation report is sent to local appraiser (LA) in step ⑦. With the radio information from SAS and software configuration from GA, the SGX enclave starts conducting radio context verification for CR in step ⑧. LA enclave sends back attestation result to GA in step ⑨.

Specifically, attestation report generated by a CR contains four parts, M_i (i.e. S_i, f_i, p_i, L_i), and all of them should be verified by the local appraiser in step ⑧. The software configuration S_i is verified by checking against a set of known device software configuration received from GA. If S_i is not on the list, then it is likely that the CR platform or application software is modified. Radio configuration verification has no known list of compliant radio configuration due to dynamic spectrum availability. $\{f_i, p_i, L_i\}$ represents frequency usage, power level and location received from ith CR and they are used to determine if the CR is compliant or not. To be more specific, the appraiser verifies if the used frequency band f_i reported by CR is the same as what is assigned by SAS. And the reported power level p_i should not exceed the maximum power allowed by SAS. Note that attestation report generation software on a CR device is protected by ARM TrustZone and can access GPS hardware to obtain trusted location of the device and report it to the appraiser. If the reported location L_i is not in the vicinity of the location where the CR request the frequency, the CR is not compliant. Time is not included in the measurement because it is implicit at the execution time of the attestation. In conclusion, CRs are audited by LA to ensure that they do not exceed the maximum transmission power at given location on assigned frequency by SAS.

B. Privacy-Preserving Multiple Devices Attestation

Running previous attestation for all CR nodes can satisfy the security requirement but it is not scalable. It leads to tremendous amount of attestation burden on IAS and energy consumption on resource constraint CR node. In addition, SAS and GA have to remote attest huge number of SGX enclaves. In the following paragraphs, we solve the scalability challenge step by step.

(1) Previous design requires BS to create one enclave for each CR device. Our first idea is a single enclave design which establishes only one LA enclave at the BS so that SAS and GA only need to perform remote attestation on one enclave for each BS. This single enclave design also reduces computation burden at BS. In this design, each device needs to carry out a remote attestation on that one LA enclave on BS before they send radio context attestation report to it. However, instantiating remote attestation is power consuming and the problem of tremendous amount of attestation burden for IAS server is not solved.

(2) To minimize the computation tasks on CR nodes and attestation burden for IAS server, we eliminate the need of performing remote attestation of LA enclave on CR nodes. We design a **trust transfer** protocol (in Sec. IV-B2) to allow the **trust establishment** (in Sec. IV-B1) between each CR device and the corresponding LA enclave through a crypto-based authentication. The idea is for the GA to transfer its trust on the LA enclave after a successful remote attestation of the LA enclave to the CR devices. This is done by GA passing necessary crypto keys to the LA enclave once it is successfully attested. This design is chosen by PriRoster to fulfill the scalability requirement.

1) Trust Establishment: We use certificate to achieve mutual authentication between CRs and the GA, and a shared secret key will be generated through SSL protocol as shown in Fig. 3. GA will carry out a remote attestation to the LA enclave with help of IAS server to ensure its integrity. GA's trust on LA enclave will be established once the attestation passes. Note that for a CR device, the keys are stored in its trusted hardware and the key related computation are performed in trusted execution environment provided by ARM Trustzone.

2) Trust Transfer: The purpose of trust transfer is to allow GA to transfer its trust on the LA enclave after a successful

remote attestation of the LA enclave to the CR devices. In this way, CRs can obtain trust on LA without doing the costly attestation. Then radio context attestation can be performed at the trusted LA confidentially and safely. Specifically, trust transfer is realized by transferring the shared secret key between CR and GA to LA enclave .

Fig. 3 shows the trust transfer from CR and global appraiser to CR and local enclave appraiser. Global appraiser generates the shared secret key (k_{ij}) with CR. If there is already an enclave for the service requested by the CR, the global appraiser will send k_{ij} to local appraiser. As mentioned above the message will be encrypted. Otherwise, the global appraiser will first negotiate with IAS server to establish an enclave in the base station and send k_{ij} to the enclave. Correct software configurations $Conf$ for the service are also sent to the enclave at the same time. However, any updates of $Conf$ will be pushed to the enclave when $Conf$ changes according to the CRN policy. The latest $Conf$ data will be stored in the local enclave appraiser for attesting the devices requesting the same service. The trust between CR and global appraiser are transferred to CR and local enclave appraiser through the secret key transfer.

Through trust establishment and trust transfer, CRs can build trust on the LA enclave without performing attestation on it. The enclave attestation is a one-time task which is done by GA. Radio context attestation can be conducted by LA securely. The system is scalable enough to handle large number of resource-constraint CR devices. Note that SAS can also use trust transfer to delegate LA enclave attestation tasks to GA and reduce the computation burden on SAS and IAS server. But for now, we consider SAS does not fully trust GA and might want to do the attestation itself.

C. Defense Against Side Channel Attack

The essence of software configuration appraisal is comparing hash of reported software configuration against hash of a set of known benign device software configuration one by one. Any software configuration which is not shown on the known list is considered malicious. If the comparison succeeds before traversing to the end of the list, the comparison process stops and put device id of the CR device under appraisal as under normal operation in final report to be sent to global appraiser. This early termination of comparison design brings side channel concerns including timing, power, memory access pattern and others. We evaluate this design in Fig.4.

However, even if we forfeit the early termination design and move to a full traversal of benign list every time design, the memory access pattern of the comparison process can still inform an attacker which software configuration is selected because the device id is accessed when a comparison matches. A one-to-one mapping of software configuration and memory access pattern can be done by controlling one CR device and observing the memory location of returned matched software configuration or by observing long term distribution pattern of attested software configurations. We show evaluation of this information leakage in the in Fig.5.

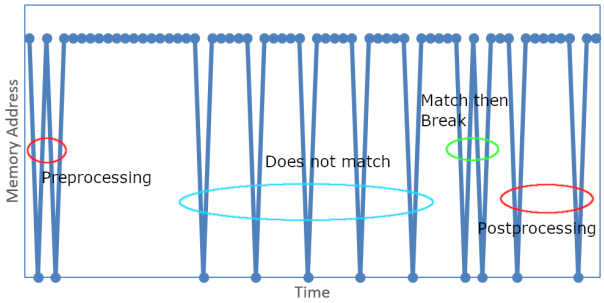

Fig. 4: Memory Access Pattern of Native Appraisal Process.

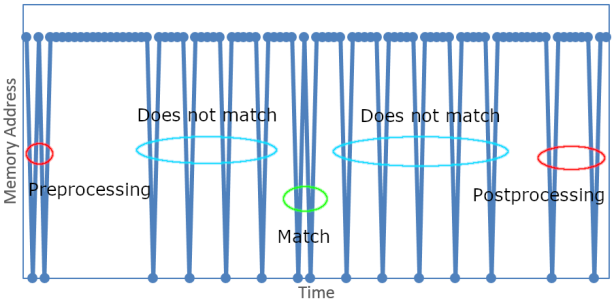

Fig. 5: Memory Access Pattern of Appraisal Process with Full Traversal Design.

To mitigate this situation, we realize oblivious software configuration appraisal by designing oblivious function with X86 cmovz instruction. X86 cmovz instruction moves the source operand to the destination operand if the condition code is true. When both source and destination operands are put in registers, this data transfer turns out to be oblivious and leaks no information about the branch selection. Our design is similar to [14], [16], [17]. We design an OCompare() function as shown in Fig. 6 to hide the trace of software configuration comparison by using cmovz instruction. This function takes in input including hash of two software configurations and return the device id only if the two hashes match. If the two configurations mismatch, this function does not change the storage for result. We elaborate the function by going through each instruction as following. The cmp instruction compares received hash of software states and update Zero flag (ZF) in EFLAGS register to reflect the comparison. The subsequent cmovz instruction copies id into the destination register according to ZF. The test instruction resets EFLAGS register by comparing known values. Fig.7 shows the process. OCompare() present the same memory access pattern since the operation is done all within registers. Therefore, an attacker can not distinguish from memory traces which software configuration is selected from memory access trace.

V. Security Analysis

In this section, we prove the secuirty of PriRoster local appraisal process in terms of radio context, compliance rules and memory oblivious function.

```
uint32_t Ocompare(uint32_t res, uint32_t sw1,
    uint32_t sw2, uint32_t id) {
    uint32_t result;
__asm__ volatile (
    "cmp %2, %3;"
    "cmovz %4, %0;"
    "test %2, %2;"
    : "=r" (result)
    : "r" (res), "r" (sw1), "r" (sw2), "r" (id)
    : "cc"
);
return result;
}
```

Fig. 6: OCompare() function. [NZ: consider removing since it is the same with previous paper]

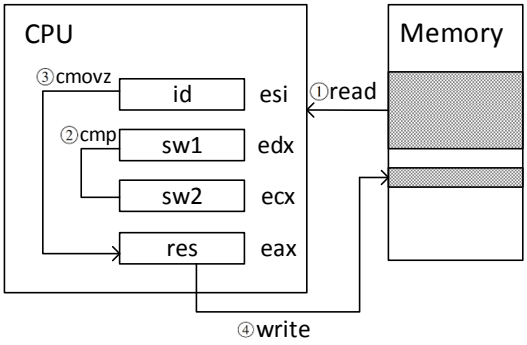

Fig. 7: OCompare() function diagram.

Theorem 1: Malicious edge BS with capability of controlling privileged software and manipulating relayed network packets cannot reveal radio context received from CR nodes in attestation report during local appraisal process.

Proof: Predistributed public secret key pair on GA and CR node guarantees a secure shared secret key generation between CR node and GA by cryptography. Edge BS cannot reveal the shared secret key through manipulating or monitoring traffic on packets between GA and CR nodes. GA remote attest LA enclave before it transmits the shared secret key to LA. If this remote attestation does not pass, GA would forfeit the process of transmitting shared secret key to LA enclave. Thus, in this case, LA enclave does not obtain the shared secret key and cannot decrypt attestation report received from CRs. Since LA enclave cannot know the attestation report content, confidentiality is protected against malicious edge BS. If LA enclave pass the remote attestation instantiated by GA, LA enclave is capable of decrypting attestation report received from CRs. However, trusted hardware (i.e. SGX enclave) guarantees the integrity and confidentiality of protected application, even if the privileged software is compromised. Thus, malicious edge BS cannot reveal contents of local appraisal process and only knows some program is at running state even if it controls operating system or hypervisor. Thus, malicious edge BS cannot gain knowledge about radio context in attestation report in this case also.

Theorem 2: Malicious edge BS with capability of controlling privileged software and network packets cannot reveal compliance rules received from SAS during local appraisal.

Proof: SAS remote attest LA enclave before it transmits compliance rules to LA enclave. If this remote attestation does not pass, SAS would forfeit the process of transmitting compliance rules to LA enclave. If this remote attestation pass, SAS would construct secure channel between itself and LA enclave to transmit compliance rules. The proof is similar to proof of theorem 1, we skip it for length of paper.

Theorem 3: PriRoster's software configuration comparison procedure is secure against adversary who observes memory traces. The adversary cannot infer which comparison matches according to observed memory traces.

Proof: We define a program's interaction with memory as a trace execution τ which records an access type (read or write), an address and some contents. We express our proof using a simulation-based technique: for each run of a software configuration comparison procedure that yields a trace τ, we show that there exists a simulator program, whose software configuration under comparison is different from the original comparison procedure, that simulates the interaction of the original comparison procedure with the memory by producing a trace τ' indistinguishable from τ. More precisely, we define indistinguishability similar to semantic security in cryptography using a game between a system that runs the comparison procedure (or the simulator) and a computationally bounded adversary that interacts with the system, observes the trace, and attempts to guess whether it interacts with the original procedure or the simulator. The comparison procedure is secure when such adversaries guess correctly with probability at most $\frac{1}{2}$ plus a negligible advantage.

To ensure security of comparison procedure, we first need to evaluate the OCompare() function in Fig. 6. Since the code operates on the processor registers only and never accesses memory, it operates within the (trusted) boundary of the sealed processor chip. As such, evaluations that involve registers only are not recorded in the trace τ, hence, any register-to-register data manipulation is secure. Then, we evaluate full traversal design with OCompare() function. Because we use a full traversal design, so different software configuration input will all go through all the OCompare() functions. Hence, we can easily simulate the program with a different software configuration input and generate a trace τ', which the adversary cannot differentiate from original trace τ. Here ends the proof.

VI. Implementation

For CR device prototype hardware setting, we select Raspberry Pi 3 as application processor and USRP N210 as baseband processor. USRP N210 has been one of the standard radio platform for CR research. For CR device software setting, we apply TrustZone to build an trusted environement for the attestation software. To be specific, we use OPTEE secure kernel [18] in the secure world and build a OPTEE Static Trusted App called ATTEST with approximately 1000 software line of code (SLOC) to serve as attestation software. We use Ubuntu 15.04 with 4.6.3 ARM 64 bit Linaro Linux kernel in normal world. The radio core device driver libUHD is the software

for controlling USRP N210. It sits in the normal world and is loaded in an address known to ATTEST at runtime. The radio parameters used by LibUHD are saved as global variables in a specific memory location known to ATTEST. Upon receiving a valid remote attestation request, ATTEST will perform SHA256 checksum of the linear memory map of libUHD and code page of Operating System kernel and embed the hash result with retrieved radio parameters inside the attestation report. We refactor openSSL 1.0.1f library for cryptographic operations and secure communication.

For edge BS, we choose Intel NUC which supports Intel SGX natively. The NUC is powered by Intel i7-6770HQ Skylake CPU with 6MB cache at 2.6 GHz and 8GB DRAM. We use ubuntu 16.04 and the local appraisal enclave is built with Intel SGX SDK v2.4. For SAS and Global appraiser, we choose AWS EC2 instance with 64 bit Ubuntu Server 18.04 LTS. According to lshw, it is using Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz and 983MiB system memory.

We implement remote attestation between CR node and LA enclave on Intel NUC, Raspberry Pi and remote attestation between GA or SAS and LA enclave on Intel NUC, AWS cloud. We register our self-signed certificate with Intel SGX remote attestation service and retrieve SPID from Intel by contacting Intel customer support. We store the private key for the self-signed certificate inside secure world of Raspberry Pi and on AWS cloud.

VII. Evaluation

In this section, we first compare the performance of three designs aforementioned to highlight the scalability of PriRoster. Then, we demonstrate the effectiveness of our oblivious design to mitigate memory access side channel.

A. Prototype Comparison

In order to compare three designs, we benchmark primitives used in them. To be specific, we benchmark instantiating remote attestation on CR node. And we benchmark instantiating remote attestation on AWS cloud. Besides, we benchmark trust transfer process of PriRoster.

1) Primitives Benchmarks: We include time.h system header and use clock() function to measure remote attestation time in source code before compilation. We measure the time for a single CR device to perform a successful remote attestation on LA enclave from connection establishment with IAS server to disconnection. It turns out the average time needed is 366.45ms for this remote attestation. We also use a AVHzY USB Power Meter Tester to supply power for Raspberry pi and collect measurement of consumed power. The collected power for performing a successful remote attestation on LA enclave for a single CR device is 0.28J in average. On the other hand, we measure the time for SAS or GA to perform a successful remote attestation on LA enclave. The average time for this remote attestation is 32.7ms. We implement trust transfer process on Raspberry Pi and AWS cloud instance using Linux socket. We evaluate the process and the outcome shows that this process takes 2.57ms in average. And the energy consumed on CR device for trust transfer is in average 0.003J. Table. I summarizes the benchmark results.

TABLE I: Bechmarks on primitives

HW	Function	Time(ms)	Energy(J)
Pi	Remote attestation	366.45	0.28
Pi	Trust Transfer	2.57	0.003
AWS	Remote attestation	32.7	-

2) Design Comparison: We focus on the computation overhead and energy consumed brought by difference between the three designs. Thus, we skip the overlapped processes like CR device attestation report generation in these designs. For simplicity of demonstration, we assume that in real life setting, there are 1500 CR devices connected to one edge BS and there exists 320,000 edge BS in USA [4]. IAS is assumed to serve clients one by one. IAS time is composed of AWS time and Pi time, since IAS participates SGX enclave attestation at both sides.

We first present the design of every CR device conducting its own remote attestation on local appraiser enclave to establish trust. In this design, there are 1500 independent enclaves exist on each edge BS, and enclaves are created or destroyed with CR's joining and leaving BS. 1500 CR nodes each need to perform attestation on one enclave. SAS and global appraiser also need to perform 1500 times of remote attestation on one enclave respectively. So SAS and GA need to perform 960,000,000 times of remote attestation in this case. And this will take 363 days for a single cloud instance. Altogether 37.33 kWh for all CR devices. And the overall processing time for IAS is 6.08 years single machine time. Furthermore, a CR needs to attest LA enclave periodically as frequently as the radio context attestation.

In the single enclave design, BS does not create a single enclave for each CR node. Instead, only one local appraiser enclave is created on BS. Thus, SAS and global appraiser only need to perform 1 time of remote attestation on this enclave respectively. So altogether the attestation time for this single BS will be around 10 minutes. SAS and GA need to perform 320,000 times of remote attestation in this case. And this will take 2.91 hours for a single cloud instance. But all CRs still need to attest the enclave which will cost 37.33 kWh. The overall processing time for IAS is 5.57 years single machine time. Similarly, CRs need to attest LA enclave periodically.

In PriRoster, each CR device do not need to remote attest local appraiser enclave but need to perform trust transfer process the first time it joins in a network. SAS and GA also only need one attestation on this enclave respectively. And the trust transfer process will take 14.28 days for a single cloud instance. The trust transfer altogether consumes 0.4 kWh for all CR devices. And the overall processing time for IAS is 5.81 hours single machine time. Note that we can easily establish multiple cloud instance for bootstrapping the attestation time.

TABLE II: Bechmarks on three Designs

Design	Pi Energy	IAS Time	SAS Time	GA Time	Single BS Time
Single device design	37.33 kWh	6.56 years	181 days	181 days	10.80 mintutes
Single enclave design	37.33 kWh	5.57 years	2.9 hours	2.9 hours	9.16 minutes
PriRoster	0.4 kWh	5.81 hour	2.9 hours	14.4 days	3.92 seconds

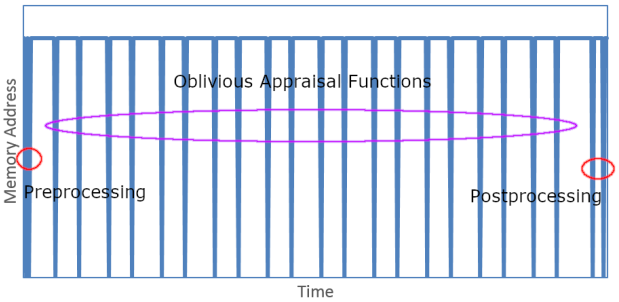

Fig. 8: Memory Access Pattern of Oblivious Appraisal Process.

B. Oblivious Appraisal Process

We show the effectiveness of oblivious appraisal function in this section. We use dynamic instrumentation tool, Intel Pin Tool 3.0 [19], for tracing memory access pattern.

We choose full traversal design to protect against side channels brought by early termination design. In addition, to hide memory access trace, we apply oblivious compare function OCompare() in Fig. 6. For every comparison, we use OCompare() to replace previous comparison function. At the end of the comparison procedure, device id is saved in result buffer if a match is found or else a dummy value will be saved in result buffer. Fig. 8 shows the oblivious appraisal process and for all matches, the memory traces stay the same. As in Fig. 8, we can see that an attacker cannot infer which software configuration is matched since all comparisons' memory trace appear to be the same.

VIII. Related Work

Although PriRoster is the first work to provide privacy-preserving radio context attestation, there has been closely related works on remote attestation, CRN security and side channels in trusted execution environment.

Remote attestation of software on a prover for a single appraiser is well studied. The prover is the device under attested and it sends a status report of its current execution state to a appraiser. Since malicious software on the prover could potentially forge the report, various methods have been proposed to promise the trustworthiness of the report. For example, [20]–[25] put secure hardware in use and [26]–[30] take advantage of trusted software. Recent interest arises on malicious actors with hardware attack capabilities also. [31], [32] take a first step to use remote attestation for protecting against hardware attacks. Besides attestation of one prover to one appraiser, [33], [34] propose swarm attestation for integrity of a group of devices. In this work, we consider remote attestation under a centralized edge computing architecture using secure hardware.

For CRN security, [35] propose authentication of CR device with signal at the physical layer and [36] propose detecting and preventing malicious CR at device level. Although authentication can verify the identity of a CR device and device level security protects a CR device from being compromised, they cannot ensure authority that every connected CR device is benign and complies to transmission permissions at runtime in our case. To ensure authority the operational integrity of the CR devices and provide insights for authority to verify their compliances, [4] comes up with remote attestation of radio context. Despite [4] provides operational integrity of CRN, the potential privacy leakage inside edge BS of the network is not considered.

Side channel information leakage on trusted system remains an active area of research [13], [14], [16], [17], [37]–[39]. [13] proposed page-fault side-channel attacks on SGX, where an attacker controlling priviledged software could extract secrets from enclave execution by tracking memory access patterns at the granularity of memory pages. [40] demonstrates another attack approach by using branch shadowing to infer the control flow of the execution inside an enclave. Branch shadowing requires requires frequently interrupting the victim enclave, and this observation enables effective detection methods [37]. [14], [16], [17] researches on information leakage of search index through memory access pattern. [38] proposes a generic path oram [39] enclave for hiding memory traces. In PriRoser, we put memory access pattern side channel under consideration and design OCompare() function for preventing information disclosure of this type.

IX. Discussion

In the current design, PriRoster only considers software attacks on CR devices. However, PriRoster can be extended to protect against physical hardware attacks by incorporating heartbeat protocols inside local enclaves similar to [31], [32]. The essence of heartbeat protocol is that if a device does not claim its existence in the network within a time interval periodically, it will be considered compromised. This is based on the observation that to facilitate a hardware attack, an attacker needs to turn off the device and thus the device won't be able to keep claiming its existence inside a network.

In our case, global appraiser can keep a list of CR devices with their status in its database. And each CR device can have 4 different status which are unseen, alive, awaiting and compromised. Unseen means a device has not been turned on for the first time, alive means a device is currently under

regular operation, awaiting means a device is either moving or under physical attack and compromised means a device is modified and no long trusted. Other than the list kept by global appraiser, local enclaves will keep a list of CR devices connected to itself with two status, compromised or alive. They will periodically ping the CR devices for evidence of staying online using a heartbeat protocol. The first time a CR device join the network, global appraiser receive a notification and change the status of the device from unseen to alive and local enclave add an entry for the device. Whenever the CR device stop its connection with an edge BS, global appraiser will receive a notification about the event and change the status of the CR device from alive to awaiting. After a predefined interval of time, global appraiser will set the status from awaiting to compromised. If the CR device move and join another edge BS within the time interval, global appraiser will receive a notification and change the status of the CR device from awaiting back to alive. Through this way, hardware attacks can be detected.

X. Conclusion

In this paper, we propose PriRoster, a privacy-preserving radio context attestation framework for CRN. PriRoster integrate trusted hardware, Intel SGX, in a scalable way to prevent information leakage at edge BS. Our design take in consideration of reducing remote attestation computation overhead and mitigating memory trace side channel of SGX enclave. We evaluate our design through implementing prototype and benchmark procedures. Through evaluation, we conclude that PriRoster is a scalable and secure framework to ensure operational integrity of future CRN.

References

[1] D. Cabric, S. M. Mishra, and R. W. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *Signals, systems and computers, 38th Asilomar Conf. on*, vol. 1, pp. 772–776, Ieee, 2004.

[2] A. Ghasemi and E. S. Sousa, "Spectrum sensing in cognitive radio networks: requirements, challenges and design trade-offs," *IEEE commun. Mag.*, vol. 46, no. 4, 2008.

[3] M. M. Sohul, M. Yao, T. Yang, and J. H. Reed, "Spectrum access system for the citizen broadband radio service," *IEEE Commun. Mag.*, vol. 53, no. 7, pp. 18–25, 2015.

[4] N. Zhang, W. Sun, W. Lou, and et al., "Roster: Radio context attestation in cognitive radio network," in *2018 IEEE CNS*, pp. 1–9, 2018.

[5] G. Coker, J. Guttman, P. Loscocco, and et al., "Principles of remote attestation," *Int. J. of Inf. Security*, vol. 10, no. 2, pp. 63–81, 2011.

[6] "Which telecoms store your data the longest? secret memo tells all." https://www.wired.com/2011/09/cellular-customer-data/.

[7] "At&t's plan to watch your web browsing—and what you can do about it." https://arstechnica.com/information-technology/2015/03/atts-plan-to-watch-your-web-browsing-and-what-you-can-do-about-it/.

[8] L. Cauley, "Nsa has massive database of americans' phone calls," *USA today*, vol. 11, no. 06, 2006.

[9] X. He, R. Jin, and H. Dai, "Camouflaging mobile primary users in database-driven cognitive radio networks," *IEEE Wireless commun. Letters*, 2018.

[10] S. Jajodia, "Adversarial and uncertain reasoning for adaptive cyber defense: Building the scientific foundation," 2015.

[11] V. Costan and S. Devadas, "Intel sgx explained.," *IACR Cryptology ePrint Archive*, vol. 2016, no. 086, pp. 1–118, 2016.

[12] F. Brasser, U. Müller, A. Dmitrienko, and et al., "Software grand exposure: Sgx cache attacks are practical," *arXiv preprint arXiv:1702.07521*, p. 33, 2017.

[13] Y. Xu, W. Cui, and M. Peinado, "Controlled-channel attacks: Deterministic side channels for untrusted operating systems," in *2015 IEEE S&P*, pp. 640–656, 2015.

[14] W. Sun, R. Zhang, W. Lou, and Y. T. Hou, "Rearguard: Secure keyword search using trusted hardware," *IEEE INFORM*, 2018.

[15] A. ARM, "Security technology building a secure system using trustzone technology (white paper)," *ARM Limited*, 2009.

[16] O. Ohrimenko, F. Schuster, C. Fournet, and et al., "Oblivious multi-party machine learning on trusted processors.," in *USENIX Security Symp.*, pp. 619–636, 2016.

[17] A. Rane, C. Lin, and M. Tiwari, "Raccoon: Closing digital side-channels through obfuscated execution.," in *USENIX Security Symp.*, pp. 431–446, 2015.

[18] "Optee." https://github.com/OP-TEE/optee_os.

[19] V. J. Reddi, A. Settle, D. A. Connors, and et al., "Pin: a binary instrumentation tool for computer architecture research and education," in *2004 workshop on Computer architecture education: held in conjunction with the 31st Int. Symp. on Computer Architecture*, p. 22, ACM, 2004.

[20] K. Eldefrawy, G. Tsudik, A. Francillon, and et al., "Smart: Secure and minimal architecture for (establishing dynamic) root of trust.," in *NDSS*, vol. 12, pp. 1–15, 2012.

[21] J. Kong, F. Koushanfar, P. K. Pendyala, and et al., "Pufatt: Embedded platform attestation based on novel processor-based pufs," in *51st Annu. Design Automation Conference*, pp. 1–6, ACM, 2014.

[22] X. Kovah, C. Kallenberg, C. Weathers, and et al., "New results for timing-based attestation," in *2012 IEEE S&P*, pp. 239–253, 2012.

[23] H. Park, D. Seo, H. Lee, and et al., "Smatt: Smart meter attestation using multiple target selection and copy-proof memory," in *Computer Science and its Applications*, pp. 875–887, Springer, 2012.

[24] S. Schulz, A.-R. Sadeghi, and C. Wachsmann, "Short paper: Lightweight remote attestation using physical functions," in *4fourth ACM Conf. on Wireless network security*, pp. 109–114, 2011.

[25] N. Zhang, K. Sun, W. Lou, and et al., "Case: Cache-assisted secure execution on arm processors," in *2016 IEEE S&P*, pp. 72–90, 2016.

[26] R. Kennell and L. H. Jamieson, "Establishing the genuinity of remote computer systems.," in *USENIX Security Symp.*, pp. 295–308, 2003.

[27] Y. Li, J. M. McCune, and A. Perrig, "Viper: verifying the integrity of peripherals' firmware," in *18th ACM CCS*, pp. 3–16, 2011.

[28] A. Seshadri, A. Perrig, L. Van Doorn, and et al., "Swatt: Software-based attestation for embedded devices," in *null*, p. 272, IEEE, 2004.

[29] A. Seshadri, M. Luk, and A. Perrig, "Sake: Software attestation for key establishment in sensor networks," in *Int. Conference on Distributed Computing in Sensor Systems*, pp. 372–385, Springer, 2008.

[30] A. Vasudevan, J. McCune, J. Newsome, and et al., "Carma: A hardware tamper-resistant isolated execution environment on commodity x86 platforms," in *7th ACM Symp. on Information, Computer and commun. Security*, pp. 48–49, 2012.

[31] A. Ibrahim, A.-R. Sadeghi, G. Tsudik, and et al., "Darpa: Device attestation resilient to physical attacks," in *9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, pp. 171–182, 2016.

[32] A. Ibrahim, "Aid : Autonomous attestation of iot devices," 2018.

[33] N. Asokan, F. Brasser, A. Ibrahim, and et al., "Seda: Scalable embedded device attestation," in *22nd ACM SIGSAC CCS*, pp. 964–975, 2015.

[34] M. Ambrosin, M. Conti, A. Ibrahim, and et al., "Sana: secure and scalable aggregate network attestation," in *2016 ACM SIGSAC CCS*, pp. 731–742, 2016.

[35] X. Jin, J. Sun, R. Zhang, and et al., "Specguard: Spectrum misuse detection in dynamic spectrum access systems," *IEEE Trans. on Mobile Computing*, 2018.

[36] Y. Dou, K. C. Zeng, Y. Yang, and et al., "Madecr: Correlation-based malware detection for cognitive radio," in *2015 IEEE INFOCOM*, pp. 639–647, 2015.

[37] M.-W. Shih, S. Lee, T. Kim, and et al., "T-sgx: Eradicating controlled-channel attacks against enclave programs," in *2017 NDSS*, 2017.

[38] S. Sasy, S. Gorbunov, and C. W. Fletcher, "Zerotrace: Oblivious memory primitives from intel sgx," in *NDSS*, 2017.

[39] E. Stefanov, M. Van Dijk, E. Shi, and et al., "Path oram: an extremely simple oblivious ram protocol," in *2013 ACM SIGSAC CCS*, pp. 299–310, ACM, 2013.

[40] S. Lee, M.-W. Shih, P. Gera, and et al., "Inferring fine-grained control flow inside sgx enclaves with branch shadowing," in *26th USENIX Security Symp.*, pp. 16–18, 2017.

ACKNOWLEDGMENT

The authors would like to thank...