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

Ruide Zhang*, Ning Wang*, Ning Zhang†, Zheng Yan‡, Wenjing Lou*, and Y. Thomas Hou*
*Department of Whatever, 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 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 a CRN heavily depends on compliance of cognitive radios (CRs). Although remote attestation of a CR's radio context is a promising solution, current remote attestation that requires the target's configuration to be publicly available to the verifier poses a fundamental challenge to the operational security of spectrum users, especially military primary users.**

To protect a device's configuration information, we propose PriRoster, a privacy-preserving remote attestation mechanism, that effectively separates the need to know the operational configuration from the capability to execute the verification process correctly at the verifier. PriRoster hides sensitive device and/or radio configuration information from untrusted intermediate verifiers in a public network and enables a range of new applications such as efficient network-wide radio context attestation. Trusted execution environment (TEE) such as Intel SGX is used in our design to provide confidential processing. However, naive application of TEE suffers from not only poor system scalability, but also information side channel leakage. We develop trust transfer protocol to significantly enhance system scalability, and the protection against information side channel attack is accomplished by automatically incorporating obliviousness primitive into the attestation program.

We build a prototype of the proposed PriRoster system using Raspberry Pi, USRP, Intel NUC, and AWS cloud. The feasibility of our proposed framework is demonstrated by system benchmarks and the effectiveness of the proposed oblivious appraisal functions are verified by recording memory access pattern via code instrumentation.

I. Introduction

With the large scale deployment of smart devices, the world has witnessed an increasing utilization of wireless communications in the last decade [1]. According to Cisco, global mobile data traffic will reach about 25 Exabytes per month by the end of 2019 [2]. 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 more efficient use of the available spectrum.

One of the key technologies to improve spectral efficiency is spectrum sharing in a cognitive radio network (CRN) [3], where opportunistic access to the radio spectrum that was originally allocated to the primary users (PUs) exclusively is now allowed to be accessed by secondary users (SUs) when the spectrum is not used by the PUs. In [4], the U.S. Federal Communications Commission (FCC) has described a dynamic spectrum management framework for a Citizen Broadband Radio Service (CBRS) governed by a spectrum access system (SAS). Based on the spectrum utilization plans from PUs and radio environment maps from sensing partners such as Google, SAS manages the use of available spectrum opportunities for SUs by granting transmission permits to CRs based on their access level and location.

While spectrum sharing holds great promises, correct operations of SAS often assume honest participation. CRs need to faithfully report the sensing results and strictly follow the transmission permits issued by the SAS. Due to the dynamic reconfigurabiity, selfish users or malicious attackers can easily reconfigure their radios to gain unfair advantage or to cause harm to the network.

One way to ensure the operational correctness is via remote attestation. Remote attestation is a process of making a claim about properties of a *target* by supplying evidence to an *appraiser* or *verifier* over a network [5]. The primary objective of remote attestation is to provide verifiable evidence about the state of software executing on a system. This evidence is intended to ensure that targets will not engage in some class of misbehavior. The process of verifying the verifiable evidence on appraiser is called appraisal. Remote attestation may be used to address a number of trust problems, including guaranteed invocation of software, delivery of premium content to trusted clients, assuaging mutual suspicion between clients, and more. In the context of CRN, remote attestation provides cryptographically verifiable evidence on the state of a CR device to prove the compliance of the CR. In [6], remote attestation is used to measure the cognitive radio context as a compliance check, however, the problem of spectrum availability privacy receive little attention. In order for the verifier to assess whether the received configuration is correct or not, he will need to possess the full knowledge of legit configurations. In CRN, this configuration consists of software configuration, radio configuration and location. To make the decision on the compliance of CR, the verifier not only need

This work was supported in part by NSF under grants CNS-1443889 and CNS-1642873, and in part by ARO under contract W911NF-18-1-0305.

978-1-7281-2376-9/19/$31.00 ©2019 IEEE

to know the CR radio context but also the full spectrum information, which is often sensitive. For example, leakage of location trajectory and transmission parameters is a serious concern for PUs that are sensitive military devices [7]. With these highly sensitive information, a malicious actor can infer where a military base is located and where an army is heading towards [8]. And leakage of software configurations can also allow an adversary to make use of known vulnerabilities in a CR device [9].

Existing approaches towards protection of spectrum information privacy is to treat the SAS as a sensitive database, and secure computation techniques are used to construct privacy-preserving queries [10]. However, direct application of these techniques can lead to significant scalability issue both in the number of radio devices and the number of possible configurations. First, the number of possible configurations for each device is not a small number since a radio context configuration consists of not only the software configuration but also the location and radio transmission parameters, which leads to many possible combinations of legit radio context configuration. Second, cryptographic privacy-preserving methods often involves significant computation overhead even if the problem can be formulated as a multi-party computation problem. Furthermore, billions of radio devices are expected to be connected to the mobile network, this shear scale would require an efficient method to handle the configuration verification in CRN attestation.

In this work, we present PRIvacy-preserving Radio cOntext atteSTation in cognitivE Radio networks (PriRoster). We achieve the goal of preserving privacy of a local appraiser (LA) on an edge base station (BS) by introducing trusted hardware, i.e. Intel SGX [11]. While building a secure system on top of Intel SGX is mostly a development effort, 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 for mutual verification between CRs and BSs. For a CR device to establish trust on an edge BS before uploading the attestation report, the CR device needs to perform remote attestation on the SGX enclave inside the edge BS. However, CR devices are resource-constrained and frequently performing remote attestation on SGX enclave consumes energy and adds unacceptable computation burden on the Intel Attestation Service (IAS). Furthermore, creating independent SGX enclaves for a large amount of CR devices introduces a large computation load on the edge BSs. In PriRoster, CR devices delegate the power-consuming attestation on SGX enclaves to the more powerful SAS server 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 to extract sensitive information. In our case, an edge BS can infer the radio context of CR devices from their memory access patterns which are observable by the edge BS. In PriRoster, we design oblivious appraisal functions for preventing memory access pattern leakage.

To summarize, our contributions are:

- We propose PriRoster, a privacy-preserving radio context attestation technique that allows a untrust verifier to carry out remote attestation of a CR device's context without knowing the device's context information itself. This technique can effectively conceal the operational parameters of the PUs' as well as the CR devices' from untrusted network components such as an intermediary edge BS.
- We consider a systematic network-wide large-scale remote attestation which allows a large number of remote devices be attested simultaneously and efficiently. We propose a novel trust transfer mechanism to address the scalability problem raised in this scenario. Individual devices can rely on the attestation result done by an trusted entity rather than each carrying out a separate attestation process.
- To address the memory side channel limitation of Intel SGX, We design an oblivious appraisal function that effectively prevents leakage of sensitive PU information through memory access at the edge BS.
- We build a prototype system of PriRoster using USRP, Raspberry Pi, Intel NUC, and Amazon AWS. The prototype system shows the feasibility of the PriRoster framework.

II. Background

A. Spectrum Sharing in CRN

To tackle the problem of spectrum scarcity, spectrum sharing is proposed to allow new entrants to utilize the radio spectrum allocated to incumbents when the spectrum is not in use. The spectrum sharing solutions can be divided into two categories: decentralized and centralized. Decentralized solutions are not reliable because of sensing challenges such as hidden node problem. The centralized dynamic spectrum management framework is increasingly attracting more attention. FCC has proposed a centralized dynamic spectrum management framework for CBRS governed by SAS. It is a three-tiered spectrum authorization framework accommodating a variety of commercial uses on a shared basis with incumbent federal and non-federal users of the 3.5 GHz band. The three tiers are: Incumbent Access(IA), Priority Access (PA), and General Authorized Access (GAA) [15]. IA has the highest priority while GAA has the lowest. The CR devices in this paper refer to the devices at PA or GAA level.

The SAS is capable of dynamic frequency assignment and interference management [16]. The core of the SAS is a database system which receives feedings from incumbent users regarding spectrum usage information, such as usage duration and operational parameters. Operational parameters include primary user identity, location, transmission power, antenna parameters, and interference tolerance. With the spectrum

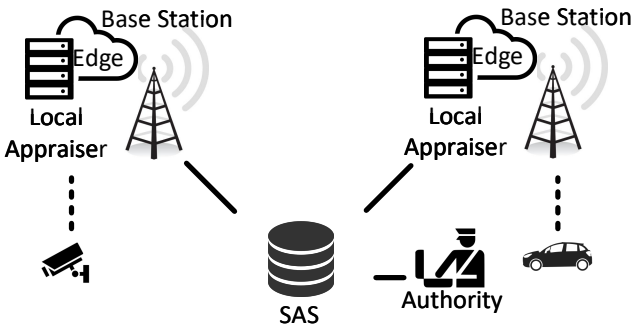


Fig. 1: Radio Context Attestation in CRN.

usage information provided, the SAS determines the available frequency within an area at a time slot and assign them to nearby CRs and determines the maximum transmission power [15], [16]. Meanwhile, SAS is responsible for detecting and removing CRs that do not obey its assignment.

B. Radio Context Attestation in CRN

The security of the SAS system involves the protection of the SAS databases and functions at the servers and the confidentiality and integrity protection of the operational CR devices in the field. In [6], we proposed a remote attestation framework for CRNs that aims to ensure the operational integrity of the CR devices by remote radio context attestation. As shown in Fig. 1, there are three major entities in the architecture- SAS, Regulatory Authority (RA) and Local Appraiser (LA). RA is a regulatory entity like FCC and LA denotes a local appraiser typically hosted on an edge base station. RA informs SAS to start attestation tasks by sending it an attestation token. Upon receiving the token, SAS delegates its appraisal tasks to LA and LA performs local appraisal of attestation reports from radio devices. In that architecture, both RA and LAs are trusted entities in the network. However, since edge base stations do not have same security level as SAS and is more likely to be compromised, the sensitive information is not safe kept on LA. Thus, in this paper, we consider the protection of sensitive information released to LAs and we integrate trusted hardware to mitigate information leakage from LAs.

C. Intel SGX

Intel SGX is Intel's latest 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 the enclaves. Intel SGX also provides remote attestation and provision, which allows a remote party like a SAS server to verify an application enclave's identity and securely provision keys, credentials, and other sensitive data to the enclave on an untrust host, such as an edge BS.

Despite the new security capabilities brought by Intel SGX, there are some known security limitations in modern Intel processors. Although Intel's autonomous memory encryption engine (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]. We integrate oblivious function to mitigate this leakage.

III. System Model and Assumptions

System Goals: PriRoster is designed to take a network-wide attestation of CR devices. The aggregated attestation report, if successfully verified, is a cryptographical proof of the compliance of all the CR nodes to the spatial-temporal sensitive radio policy. During this process, the radio context of individual CRs should not be accessible by BSs, and neither should the BSs learn the full details of the PU's operational parameters.

Threat Model: For CR devices, we assume attackers can gain control of a CR device by conducting software attacks. They can thus modify radio related parameters like transmission power, modulation method and more. Attackers can also fabricate network packets coming out of the controlled device. We do not consider hardware attacks. For 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 intercept or fabricate information in and out the edge BS via its network interface. We assume an adversary can use privileged software to observe fine-grained memory trace.

Assumptions: We assume CRs are equipped with trusted hardware components like widely available ARM TrustZone [17]. We assume CRs' software stack contains normal world and secure world. And the integrity of secure world software is guaranteed by secure boot. We assume certificates of SAS and RA are preloaded to the secure world of CR nodes and certificates of both RA and CR nodes are available to SAS. 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 edge BSs, we assume they are equipped with Intel SGX [11]. We assume edge BSs can control the privileged software like hypervisor and operating system but cannot modify hardware.

IV. PriRoster Framework

PriRoster is 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 is delegated to edge BSs while only aggregated attestation results are sent back to SAS. However, radio context (location, spectrum usage, power level, operating time and software configuration) of CR node contains sensitive information. Thus, local appraisal should not leak 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 of primary users like location of military radios. Therefore, in our design, we target at preventing both CR's radio context and compliance rules in local appraisal from being leaked to edge BS. To achieve this goal, are three major challenges:

- Conducting local appraisal at untrusted edge nodes may leak sensitive information including radio context and compliance rules. To provide privacy-preserving radio context attestation, we implement LA's functionalities in an enclave on the edge BS. This process is detailed in Sec. IV-A.
- To scale up, multiple devices with same service request are assigned to share one enclave at a BS. However, remote attestation of the LA enclave needs to be conducted by each CR device to establish the trust on the LA enclave by the CR devices. This leads to non-negligible energy consumption at each CR and a tremendous amount of attestation burden on IAS server. We propose a trust transfer design which delegate the task of remote attestation of LA enclave from CRs to SAS thus minimize the number of remote attestations that need to be done in Sec. IV-B.
- Intel SGX provides confidentiality and integrity for enclave programs, however, there are known security limitations of Intel SGX itself. For example, although privileged software cannot access enclave memory, it can be used to observe memory access pattern [13]. Therefore, an attacker controlling privileged software can potentially disclose 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 Single Device Attestation

In this section, we present privacy-preserving remote attestation of radio context on a single device. We take advantage of trusted hardware (i.e. Intel SGX enclave) for defending against compromised edge BS. From a high level view, SAS distributes radio context attestation request and LA enclaves conduct the radio context attestation on behalf of SAS. Before delegating radio context attestation task to LA enclave, SAS needs to assess the trustworthiness of LA enclave's execution environment by performing remote attestation on it. Similarly, CR node needs to first verify the trustworthiness of LA enclave before accepting the attestation request from it. Then CR sends its radio context report to correctly verified LA enclave with confidence that both the integrity and confidentiality are guaranteed. In the end, LA enclave sends local appraisal results to SAS and single device attestation is completed.

As shown in Fig. 2, authority initiates a radio context attestation. SAS pushes a remote attestation request to BSs in step ① . Each BS forwards the request to the CRs within its range in step ② . Upon receiving the request, a CR in turn requests to attest the execution environment of the LA enclave running on the BS in step ③ . LA enclave replies with its

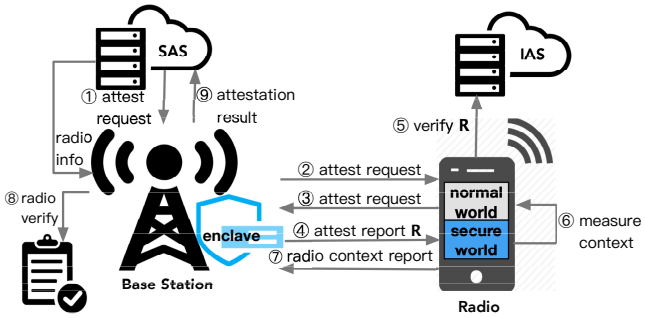


Fig. 2: Privacy-Preserving Device Attestation

enclave attestation report R to the CR node in step ④ . With the help of IAS, the CR assesses enclave's trustworthiness in step ⑤ . Only if a positive verification response from IAS is received, will the CR start radio context measurement in step ⑥ . Then the attestation report is sent to LA in step ⑦ . With the information regarding compliance rules (radio assignment information and correct software configuration) received from SAS, LA enclave conducts radio context verification for the CR in step ⑧ . In the end, LA enclave sends back attestation result to SAS in step ⑨ .

The detail of radio context attestation protocol is outlined in Fig. 3, describing a successful protocol run. Note that we assume SAS and CR nodes know the public key of RA, and RA and CRs also know the public key of SAS, as described in our assumptions in Sec. III. In addition, SAS has to set up a LA enclave on each untrusted edge node involved with the help of IAS before delegating radio context attestation task to it. After successfully setting up the LA enclave, an unique attestation key used to produce signature will be burned into each newly established LA enclave. SAS conducts authentication on LA enclave by verifying LA enclave's signature against an endorsement certificate created by manufacturer Intel. A secure channel between SAS and LA enclave will be established after LA enclave is successfully set up. Acronyms and parameters definition are shown in Table. I.

Steps ① and ② show the propogation of radio context attestation request from SAS to CR devices. After mutual authentication with RA, SAS obtains a valid token τ from RA. SAS sends the attestation request consisting of τ and a nonce N_A to local enclaves. Nonce N_A is used to resist the replay attack and to associate an attestation request with the corresponding attestation report. It can prevent an adversary from reusing old attestation requests, thus stopping potential DoS attacks where an adversary spams attestation requests on the network.

Steps ③ to ⑤ describe attestation of the LA enclave. Upon receiving a radio context attestation request, the CR node verifies the token generated by RA and check the included nonce N_A to ensure the freshness of this request. If the request is verified correctly, CR node initializes a request to attest the

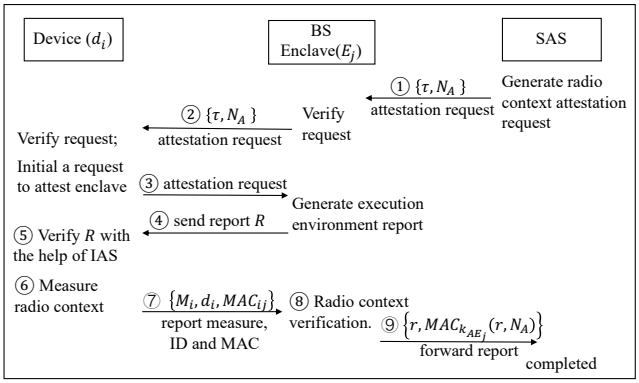


Fig. 3: The radio context attestation protocol

TABLE I: Acronyms & Parameter Definition

RA	Regulatory authority
SAS	Spectrum access system
LA	Local appraiser
IAS	Intel Attestation Service
k_{ij}	Shared secret key between CR d_i and local enclave appraiser E_j
k_{AE_j}	Shared key between global appraiser and base station enclave E_j
$\hat{S}_i$	Measured software configuration of d_i
$\hat{f}_i$	Measured frequency band used by d_i
$\hat{p}_i$	Measured power level of d_i
$\hat{L}_i$	Location measurement of d_i
$Conf$	Correct software configuration at SAS
τ	Attestation token from RA
N_A	Nonce generated by RA for attestation
d_i	Identification of CR device i
MAC_{ij}	MAC generated by d_i using key k_{ij}

execution environment of LA enclave. This verification is done with the help of IAS, and detail of SGX enclave attestation can be found in [18].

Steps ⑥ to ⑨ are the radio context measurement and report process. Radio context M_i is measured by the attestation routine inside ARM TrustZone of CR device. A CR device i then generates the response $\{M_i, d_i, MAC_{ij}\}$, where $MAC_{ij} = MAC(M_i, d_i, N_A)$, using the shared secret key k_{ij} between CR device i and LA enclave j. MAC value is used to ensure both source and content integrity of the report. M_i, the radio context, contains four parts, $\{\hat{S}_i, \hat{f}_i, \hat{p}_i, \hat{L}_i\}$, which will be explained in step ⑧ , verification of radio context, as follows.

The software configuration $\hat{S}_i$ generated by hashing the memory pages is verified by checking against a set of known benign device software configurations received from SAS. If $\hat{S}_i$ is not on the list, then it is likely that the CR platform software stack is compromised. However, there is no known list of compliant radio configurations due to dynamic spectrum availability. To verify the radio configuration, LA enclave first verifies if the used channel $\hat{f}_i$ reported by CR is the same as what is assigned by SAS. Then the power level $\hat{p}_i$ is compared with the maximum power allowed by SAS. In conclusion, CRs

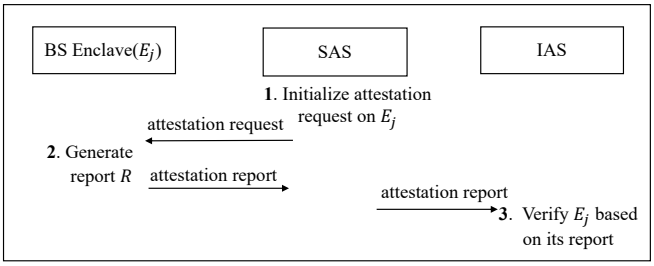


Fig. 4: Trust Establishment of SAS on SGX enclave

are audited by LA enclave to ensure that they do not exceed the maximum transmission power at given location on assigned channel by SAS. In the end, LA sends the attestation result r of a CR with corresponding $MAC(r, N_A)$ to SAS in Step ⑨ .

B. Privacy-Preserving Multiple Devices Attestation

Running single device attestation described in Sec. IV-A can satisfy the security requirement but it is not scalable. If one has to set up an LA enclave for each CR device, a large number of enclaves will have to be established which is a big burden for the host. In our PriRoster design, only one LA enclave is established at the edge BS node, and this one LA enclave will serve multiple CRs associated to this BS.

Another scalability concern is that, by the naive design, each CR device needs to carry out a remote attestation on the LA enclave it associates before it sends radio context report to the enclave. This would be duplicated efforts if multiple CRs are connected to a same LA enclave. Considering that a remote attestation is a much more expensive process comparing to a cryptographic authentication, in our PriRoster design, we release the resource-constrained CR deviced from the burden of carrying out the remote attestation of the LA enclave. Instead, we delegate the attestation of the LA enclave to the more resourceful SAS and transfer the trust established on the LA enclave by SAS to each individual CRs through an authentication protocol.

The task delegation is a two-step process: Trust Establishment and Trust Transfer.

Trust Establishment: Fig. 4 shows the trust establishment on LA enclave by SAS through conducting remote attestation on enclaves. SAS first initializes a remote attestation request on enclave E_j to assess the execution environment trustworthiness of local enclave. Local enclave E_j generates a report and sends it back to SAS. Once the attestation result is verified correctly by SAS with the help of IAS, SAS's trust on LA enclave will be established.

Trust Transfer: Following trust establishment, SAS can transfer its trust on a LA enclave to individual CRs assocaiated with that LA, through authentication protocol. We propose two implementations of trust transfer: i) Symmetric Key Transfer, ii) Public Key Certificate Distribution. Essentially, the task of attesting the trustworthiness of enclave is delegated to SAS.

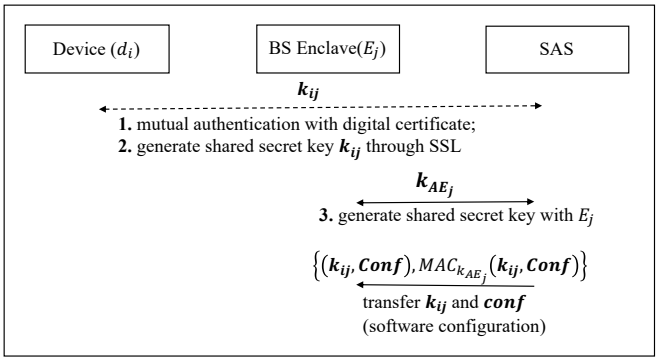
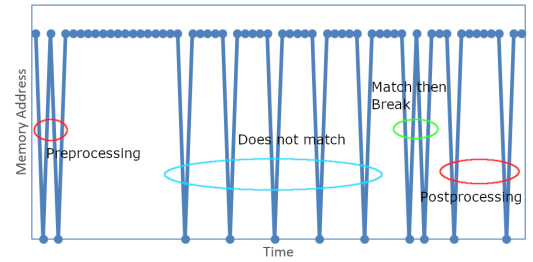


Fig. 5: Trust transfer procedure by transferring symmetric key.

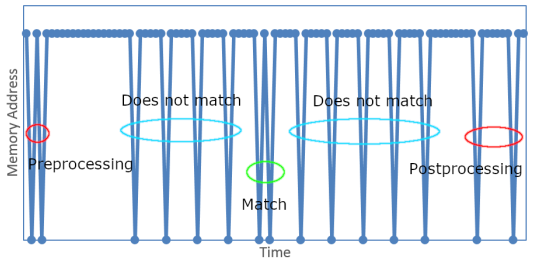
i) Symmetric Key Transfer: Trust transfer through transferring symmetric key is outlined in Fig. 5. Both SAS and CR devices have their own public keys so mutual authentication can be done between SAS and any CR i, and a shared secret key k_{ij} can be generated securely during this process, where j denotes the BS that the CR is associated with. SAS then securely transmits this shared secret key k_{ij} to LA enclave j. For a CR device, the keys are stored in its trusted hardware and cryptographic computations are performed in its secure world. Instead of carrying out a remote attestation on LA j, CR i now relies on authentication of LA j based on the shared secret k_{ij} in order to gain trust on LA enclave j.

ii) Public Key Certificate Distribution Alternatively, SAS can issue a certificate with an expiration time to an LA enclave once a successful attestation is done. LA enclave sends both the attestation request and its signed certificate to the CRs to start radio context attestation at each individual CR device. CRs establish trust on LA enclave by verifying the received certificate. In the end, CRs send back the radio context report to trusted LA. However, this method requires constant certificate verification on the CR side. And this is not suitable for defending against hardware attack. Thus, we choose the symmetric key transfer scheme.

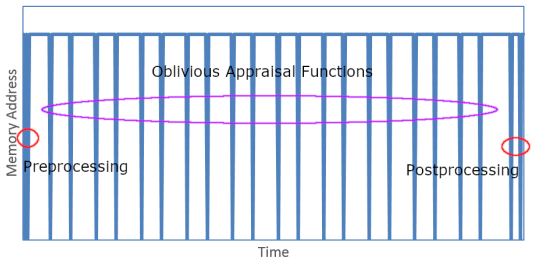
Note that authentication and attestation establish different levels of trust. Crypto authentication protocols only verify the keying material. As long as the party being authenticated demonstrates the knowledge of the secret keying material, the trust is established. However, enclave attestation verifies not only the keys, but also the code and data integrity inside the enclave. Authentication can only ensure that the party holds the right key, while attestation can also ensure the operational integrity of the party. Therefore, the trust transfer is not at the same trust level. The transfer would remain at the same level if the following assumption holds: no successful attack to the enclave between the SAS attestation and the CR authentication. We made this assumption as it is very likely to be true and the delegation of attestation tasks allows significant computation savings in the overall system.



(a) Memory Access Pattern of Naive Appraisal Process.



(b) Memory Access Pattern of Appraisal Process with Full Traversal Design.



(c) Memory Access Pattern of Oblivious Appraisal Process.

Fig. 6: Memory Access Pattern of Native Appraisal Process (a), Full Traversal Design (b), and Oblivious Appraisal (c).

C. Defense Against Side Channel Attack

One of the primary tasks in software configuration appraisal is the verification of the cryptographic hash of the system memory that captures the software configuration. If the hash checksum does not match any of the known good configurations, then the device is considered compromised. However, if a matched is found before reaching the end of lists of legitimate configurations, the function returns without doing further comparisons. However, such early termination of comparison leaks side channel information allowing the attacker to extract the software configurations of the target under attestation. We perform a experiment to demonstrate the side channel information leakage of this design in Fig. 6(a).

While an enforced full traversal design would solve the early termination of hash comparison, the attacker can also exploit memory access pattern on the preparation of the result network packet. More specifically, he can observe if the attestation pass or fail based on if the real device id memory is loaded or the stub id memory is loaded. We show evaluation of this information leakage in Fig. 6(b). In comparison, with

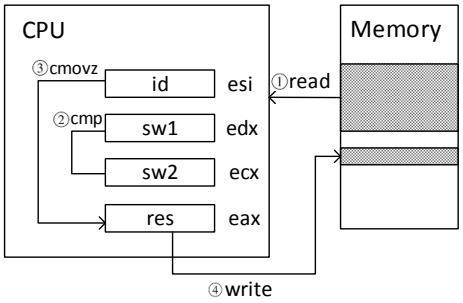


Fig. 7: OCompare() function diagram.

integration of the following oblivious function, we design an oblivious appraisal process whose memory trajectory is shown in Fig. 6(c). The detail design is discussed in Sec. VII-B.

To mitigate this information leakage, we implemented an oblivious software configuration appraisal by designing oblivious function with X86 cmovz instruction. X86 cmovz instruction moves source operand to destination operand if 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], [19], [20]. An OCompare() function is used 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. Note that, the hashes here are trimmed to fit in register. The authors consider trimmed hash is robust enough for current circumstance. If the two configurations mismatch, this function does not change the return buffer for result. The function has four main steps, 1) both values are loaded into register, 2) the cmp instruction compares received hash of software states and update Zero Flag (ZF) in EFLAGS register to reflect the comparison, 3) the cmovz instruction copies id into the destination register according to ZF, 4) the test instruction resets EFLAGS register by comparing known values. Fig. 7 shows the process. OCompare() presents 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.

V. Security Analysis

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

a) Confidentiality of the Radio Contexts and Spectrum: One of the primary security goals of PriRoser is to ensure the confidentiality of configurations of the prover (CR) from verifier (BS). There are two aspects of confidentiality in the attestation process, the confidentiality of individual provers (CRs) and the set of legal configurations derived from the sensitive spectrum information. The individual prover's configurations are protected via either remote attestation or trust verification in the transfer process. More precisely, with remote attestation, the CR can verify not only the identity but also the configuration of the system that processes his submitted information. As a result, the information is protected by the TEE in BS. Through the trust transfer process, individual CRs leverage verification of authentication token to alleviate the process of the remote attestation to the trust on authority in that he has performed the attestation and have verify the environment appropriately. For the spectrum availability, since all the information are processed within the TEE and is only used to perform attestation, its protections will be based on the security guarantee of the TEE.

b) Defense against Side Channel: We define a program's interaction with memory as a trace execution τ which records the access type (read or write) and address of 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 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 to observe 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. 7. 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, we consider any register-to-register data manipulation secure. As such, we evaluate full traversal design with OCompare() function. Since we use a full traversal design, different software configuration input will all go through all the OCompare() functions. Simulation of the program with a different software configuration as input cannot be differentiated from original trace τ by the adversary.

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 a trusted environement for the attestation software. To be specific, we use OPTEE secure kernel [21] 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, 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 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

Our evaluation of the proposed system focus on two main aspect - scalability of in large radio network context attestation, and the ability to protect confidential configuration information against side channel leakage of the TEE during the verification process.

A. Prototype Comparison

In order to effectively compare three designs, we individually benchmark the primitives used in the protocols. To be specific, we benchmark instantiating remote attestation on CR node, instantiating remote attestation on AWS cloud, trust establishment and trust transfer process of PriRoster.

1) Primitives Benchmarks: 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 consumption for performing a successful remote attestation on LA enclave for a single CR device is 0.28J on average. On the other hand, we measure the time for SAS to perform a successful remote attestation on LA enclave. The average time for this remote attestation is 32.7ms. We implement trust establishment and 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 on average. And the energy consumed on CR device for trust transfer is on average 0.003J. Table. II summarizes the benchmark results for primitives.

TABLE II: Primitive Benchmark

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 Benchmark Comparison: We focus on computation overhead and energy consumed brought by difference between the three designs of prototypes. Thus, we skip overlapped processes like radio context attestation report generation on CR nodes in these designs. For simplicity of demonstration, we assume that in real life setting, there are 1,500 CR devices connected to one edge BS and there exists 320,000 edge BS in the U.S. [6]. IAS server is assumed to serve clients one by one. We assume IAS time is composed of AWS time and Pi time, since IAS participates SGX enclave attestation in both cases.

We first present the design of every CR device conducting its own remote attestation on LA enclave to establish trust. In this single device design, there are 1,500 independent enclaves existing on each edge BS, and enclaves are created or destroyed with CR's joining and leaving BS. Therefore, CRs need to attest LA enclaves per radio context attestation request. For simplicity of comparison, we assume all CR nodes are static for now. LA enclave attestation consumes 37.33 kWh for all CR devices under all BSs. SAS need to perform 960,000,000 times of remote attestation which takes 363 days for a single cloud instance. Task at a single BS including enclave attestation by CRs, SAS takes around 11 minutes. And the overall processing time for IAS is 6.56 years of single machine time.

In the single enclave design, only one LA enclave is created on a BS for 1,500 CRs. Thus, SAS only needs to perform one time of remote attestation on this enclave respectively. But all CRs still need to attest enclaves. So altogether the attestation time for single BS is around 9 minutes. Similar to single device design, CRs need to attest LA enclave per radio context attestation request. SAS needs to perform 320,000 times of remote attestation, which takes 2.91 hours for a single cloud instance. The overall processing time for IAS is 5.57 years of single machine time.

In PriRoster, each CR device does not need to remote attest LA enclave but it needs to perform trust establishment and trust transfer process the first time it joins in a network. SAS only needs one attestation on this enclave respectively. Enclave attestation (by SAS) together with trust transfer at a single BS takes around 3.92s. The trust establishment and trust transfer process of all CRs at SAS takes 14.28 days and cost 0.4 kWh for a single cloud instance. Note that, the trust establishment and trust transfer process only takes place at CRs's joining time, so the runtime burden for SAS will be much lighter. The overall processing time for IAS server is 5.81 hours of single machine time. Note that we can easily establish multiple cloud instances and use multiprocessing

TABLE III: Design Benchmark Comparison

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

p is the percentage of CRs that join a new BS per unit time

for bootstrapping the attestation time. Suppose we have 16 threads on one server, this process only takes 20min. Table. II summarizes the benchmark results for design comparisons.

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 [22], 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(). 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 stub value will be saved in result buffer. Fig. 6(c) shows the oblivious appraisal process and for all matches, the memory traces stay the same. As in Fig. 6(c), 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 an 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, [23]–[28] put secure hardware in use and [29]–[33] take advantage of trusted software. Recent interest arises on malicious actors with hardware attack capabilities also. [34], [35] take a first step to use remote attestation for protecting against hardware attacks. Besides attestation of one prover to one appraiser, [36], [37] 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, [38]–[41] propose authentication of CR device with signal at the physical layer and [42], [43] 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, [6] comes up with remote attestation of radio context. Despite [6] 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], [19], [20], [44]–[48]. [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. [49] demonstrates another attack approach by using branch shadowing to infer the control flow of the execution inside an enclave. Branch shadowing requires frequently interrupting the victim enclave and this observation enables effective detection methods [44], [47]. [14], [19], [20] research on information leakage of search index through memory access pattern. [45] proposes a generic path ORAM [46] 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. Conclusion

In this paper, we propose PriRoster, a privacy-preserving radio context attestation framework for CRN. PriRoster integrates trusted hardware, Intel SGX, to prevent information leakage at edge BS. Our system has two key innovations. To solve the scalability challenge in remote attestation of a large network, we design a novel trust transfer protocol to allow an effective trade-off between security guarantee and scalability. To address the side-channel information leakage at the TEE, we design an input oblivious algorithm to enable radio context verification without leaking memory access information. A prototype of PriRoster is implemented to demonstrate the feasibility of the system in terms of computation, energy overhead, as well as the memory access pattern.

References

[1] "From radiotelegraphy to worldwide wireless: How itu processes and regulations have helped shape the modern world of radiocommunications." https://www.itu.int/itunews/manager/display.asp?lang=en&year=2006&issue=03&ipage=radiotelegraphy&ext=html.

[2] "Optimising leds for wireless communication." https://compoundsemiconductor.net/article/99050/Optimising_LEDs_for_wireless_communication/feature.

[3] 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.

[4] 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.

[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] 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.

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

[8] "He strava heat map and the end of secrets." https://www.wired.com/story/strava-heat-map-military-bases-fitness-trackers-privacy/.

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

[10] B. Bahrak, S. Bhattarai, A. Ullah, J.-M. J. Park, J. Reed, and D. Gurney, "Protecting the primary users' operational privacy in spectrum sharing," in *2014 IEEE International Symposium on Dynamic Spectrum Access Networks (DYSPAN)*, pp. 236–247, IEEE, 2014.

[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] M. Palola, M. Höyhtyä, P. Aho, M. Mustonen, T. Kippola, M. Heikkilä, S. Yrjola, V. Hartikainen, L. Tudose, A. Kivinen, R. Ekman, J. Hallio, J. Paavola, M. Mäkeläinen, and T. Hänninen, "Field trial of the 3.5 ghz citizens broadband radio service governed by a spectrum access system (sas)," 03 2017.

[16] M. M. Sohul, M. Yao, T. Yang, and J. H. Reed, "Spectrum access system for the citizen broadband radio service," *IEEE Communications Magazine*, vol. 53, pp. 18–25, July 2015.

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

[18] T. Knauth, M. Steiner, S. Chakrabarti, L. Lei, C. Xing, and M. Vij, "Integrating remote attestation with transport layer security," *arXiv preprint arXiv:1801.05863*, 2018.

[19] 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.

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

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

[22] 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.

[23] 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.

[24] 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.

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

[26] 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.

[27] 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.

[28] 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.

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

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

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

[32] 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.

[33] 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.

[34] 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.

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

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

[37] 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.

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

[39] X. Jin, J. Sun, R. Zhang, and Y. Zhang, "Safedsa: Safeguard dynamic spectrum access against fake secondary users," in *22nd ACM SIGSAC CCS*, pp. 304–315, ACM, 2015.

[40] V. Kumar, J.-M. Park, and K. Bian, "Blind transmitter authentication for spectrum security and enforcement," in *2014 ACM SIGSAC CCS*, pp. 787–798, ACM, 2014.

[41] Y. Liu, P. Ning, and H. Dai, "Authenticating primary users' signals in cognitive radio networks via integrated cryptographic and wireless link signatures," in *2010 IEEE S&P*, pp. 286–301, 2010.

[42] 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.

[43] C. Li, A. Raghunathan, and N. K. Jha, "An architecture for secure software defined radio," in *Conference on Design, Automation and Test in Europe*, pp. 448–453, 2009.

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

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

[46] 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.

[47] S. Chen, X. Zhang, M. K. Reiter, and Y. Zhang, "Detecting privileged side-channel attacks in shielded execution with déjá vu," in *2017 ACM on Asia CCS*, pp. 7–18, 2017.

[48] N. Zhang, K. Sun, D. Shands, W. Lou, and Y. T. Hou, "Truspy: Cache side-channel information leakage from the secure world on arm devices.," *IACR Cryptology ePrint Archive*, vol. 2016, p. 980, 2016.

[49] 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.