ROSTER: Radio Context Attestation in Cognitive Radio Network

Ning Zhang*, Wenhai Sun*, Wenjing Lou*, Y. Thomas Hou*, Wade Trappe†
* Virginia Polytechnic Institute and State University, VA
† Rugters University, NJ

***Abstract*—In cognitive radio network, cognitive radios dynamically reconfigure themselves based on the spectrum opportunity. It is envisioned to be the key of overcoming the spectrum shortage. However, such reconfigurability also amplifies potential harmful interferences from non-compliant radios.**

In this paper, we propose ROSTER, a radio context attestation protocol for cognitive radio network. The proposed protocol is based on our observation that the compliance of a radio transmission depends on software configuration, radio configuration as well as the location and time of the device, which we call *radio context*. We believe radio context attestation, which allows the authority to verify the operational integrity of individual cognitive radio, is a fundamental security function for cognitive radio networks. To the best of our knowledge, we are the first to study this important problem. Different from conventional software attestation, ROSTER is designed to handle dynamic configurations in cognitive radios. Furthermore, special considerations are given in the protocol design to accommodate different levels of sensitivity in spectrum databases. Besides protocol design and security analysis, we also build a prototype of the proposed system using Raspberry Pi, USRP, and Amazon AWS. Network simulation using the benchmark measurements from the prototype shows the scalability of our proposed protocol.

I. Introduction

With the proliferation of millions of connected smart devices, the spectrum is becoming increasingly saturated. Cognitive radio network (CRN), which provides opportunistic access to unused spectrum, is poised to become the next generation wireless communication. The vision of cognitive radio (CR) extends further beyond dynamic spectrum access. CR is envisioned to be capable of dynamically reconfiguring its radio system based on the application context. In CRN, a secondary CR can transmit on the spectrum used by primary user when there is no harmful interference to the incumbent. Such spectrum opportunity can be discovered through distributed sensing or a centrally managed spectrum database. Spectrum access system (SAS) was first described in the federal communications commission (FCC) report on citizen broadband radio service (CBRS) [1]. Using the radio environment map collected by sensing partners such as Google, SAS grants transmission permits based on the user identity and her location [1].

However, the proper operation of SAS relies on accurate report of CR status. Unfortunately, self-reported device information might not always be reliable. This imprecision could be originated from simple user misconfiguration of radio parameters or deliberate modification of the radio software to gain unfair advantage. For example, *selfish users* may attempt to misreport their identities or locations to gain transmission permits with higher bandwidth than otherwise allowed. A *malicious attacker* that controls a CR bot net, can launch a spectrum denial of service (DoS) attack by having all the zombie nodes request for spectrum access in the same geographic area. From the perspective of the primary user, repeated queries can leak sensitive operational information. Exposing location trajectory and transmission parameters through SAS queries can be a serious concern for military operations. One of the key unaddressed issues that enable these attacks is the inability to reliably capture the contexts of CRs. We observe that *the compliance of a radio transmission cannot be determined without its full context including software configuration, radio configuration, device location and time*. Collectively, we call these the ***Radio Context***.

Previous efforts on CRN monitoring focused primarily on authentication of signal at the physical layer [2], [3], [4], [5], [6]. Cryptographic spectrum permits are embedded in the physical signal such that dedicated network monitor or participants can verify the authenticity. While these systems are effective in identifying the transmitter of a signal, it does not provide the software context of the device. There are also efforts in building device level security enhancements to detect [7] and prevent [8] malicious CR. However, it is often difficult to determine the compliance of the radio without complete information of the spectrum availability.

In this work, we present Radio cOntext atteSTation in cognitivE Radio network (ROSTER). To the best of our knowledge, this is the first work to provide such fundamental security function for cognitive radio networks. Remote attestation is the process of making a claim about properties of a target by having a prover supply evidence to a verifier over network. Unlike conventional software remote attestation that examines only the software properties of a device, ROSTER aims to verify the radio context of CR, which is essential to proper operation of CRN. Through the radio context attestation, authority can gain insights into the operational integrity of the devices and verify their compliances. This capability is instrumental in network-wide policy enforcement and risk mitigation. While remote software attestation is a relatively mature field [9], [10], [11], the protocol design of ROSTER is challenged by the dynamic nature of radio context.

The first challenge is the lack of precise definition of compliant configuration. In software remote attestation, the checksum of the device software is well defined. However, radio context of a CR can be compliant at one time and location, and non-compliant at another. The context can only be audited by those with full knowledge of the spectrum policy. This dynamic nature makes naive adoption of remote

attestation infeasible. In ROSTER, each component of the radio context is audited at different steps of the protocol by different entities.

The second challenge is the sensitivity of the spectrum information. When the SAS database is sensitive, the radio context auditing is performed only by the SAS. ROSTER offers two methods for radio context checking based on the sensitivity level of the SAS.

The third challenge in our protocol design is heterogeneous device measurements. Different from software configuration, radio context signatures are different for individual CR, therefore it is infeasible to aggregate signatures of different devices and still be able to verify them efficiently [11]. In ROSTER, the base station is used to aggregate attestation results. Furthermore, to more effectively conserve energy used in CR, ROSTER adopts a symmetric key based system.

To summarize, our contributions are

- We design ROSTER, the first proposal for radio context attestation in cognitive radio network. The radio context includes software configuration, radio configuration as well as location and time. Furthermore, SAS system is leveraged to tackle the challenge of highly dynamic context in cognitive radio network. ROSTER represents the first step in a new line of research to measure the operational integrity of cognitive radios in cognitive radio network.
- We build a prototype using USRP, Raspberry Pi and Amazon AWS. The prototype of the system demonstrates the feasibility of adopting the protocol on low-end devices. ROSTER is evaluated through system benchmark and network simulation. Our benchmark on the prototype includes not only computation and network cost but also energy cost on the radio. Our simulation shows the proposed protocol scales well to large number of cognitive radios.

II. Radio Context Attestation

A. Radio Context

Proper operation of a cognitive radio relies not only on a trusted software environment, but also the intended radio configuration for the geo-location of the user in a range of time. The spatial-temporal radio context governs the essence of radio transmission compliance. Therefore, the ability to measure radio context of individual device would be one of the cornerstone capabilities for secure cognitive radio networks.

More concretely, the radio context of a CR contains four items. The first one is the software configuration S. The software configuration measures not only the integrity of the software running on the platform but also the user and application context. This is necessary because transmission rules can be different based on the type of application executing on the radio handset. A CR that is used by fire fighter in emergency response shall be prioritized in scheduling and has less restriction than a graduate student streaming video on his bus trip to the lab. The second item is the radio configuration R. It consists of the radio transmission related parameters, such as the frequency band, modulation mode, transmission power and checksum of the radio FPGA in use. The last two items are the location of the radio device and time.

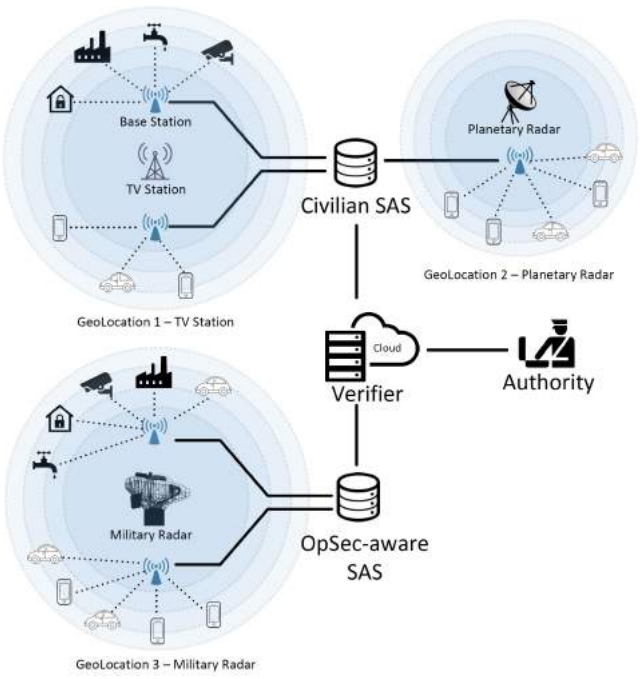

Fig. 1. Radio Context Attestation

B. System Model

In this work, we focus on cognitive radio network with centralized spectrum management, which was described in the CBRS whitepaper by the FCC [1]. An architectural level system diagram of CRN is shown in Fig. 1. Within the CRN, CRs are connected to the base station via wireless network. SAS serves as the spectrum administrator for a broad geographic region. CRs residing in different locations have their corresponding spectrum opportunities due to different primary users in vicinity. Base station and CR need to seek permission from the SAS to transmit as the secondary user. We further make the distinction of two different types of SAS, the civilian SAS and Operational Security(opsec)-aware SAS. Spectrum usage of the primary users in the civilian SAS can be disclosed to the general public. On the other hand, opsec-aware SAS is for regions where military can be one of the primary users. Operational details in these SAS cannot be disclosed. The civilian SAS can be operated by commercial companies such as Google and Microsoft, while the opsec-aware SAS can be hosted on private cloud or by trusted third party. We call the regulatory entity, such as FCC, the authority. The verifier running on the cloud can be the regulatory entity or other outsourced companies contracted to perform radio context attestation.

C. Threat Model and Assumptions

1) Threat Model: We assume malicious attackers can exploit vulnerabilities on the radio to gain control of the device. An attacker could reload the radio core to use different modulations or different parameters such as backoff window

and transmission power. An attacker may falsify her location. She may also attempt to fabricate or replay network packets to impersonate either other radios or even the verifier. We do not consider physical attacks on the CR. The physical protection of the radio can be realized via physical security measures or tamper-resistant hardware design [12].

2) Assumptions: We assume CR is powerful enough to perform light weight cryptographic functions, because it has needs for dynamic spectrum access. We assume the CR platforms are equipped with the hardware necessary to perform remote attestation, such as the widely available ARM TrustZone [13]. We assume base stations and SAS systems have adequate protection from both remote software attacks and physical attacks, thus can be trusted. We assume the communication channels between network nodes are encrypted. We also assume that base station has a fixed well known location or is equipped with location acquisition devices such as a GPS module so its location can be reliably acquired. We assume it is equipped with technology such as directional antenna to estimate the approximate locations of the connected radios.

D. System Design Goals

The primary objective is to provide the much-needed radio context attestation. The security goals include,

- *Unforgeability* - If none of the base stations or SASs is compromised, and attestation hardware is unchanged, the results reported from the trusted container in the radio to the verifier shall reflect the true configuration of the device.
- *Completeness* - If none of the base stations or SASs is compromised, and the verifier is able to validate the report signatures, then the report contains results from all the attested devices.
- *Mutual Authentication* - The proving device shall be able to verify the authenticity of the attestation request. The verifier shall also be able to verify the authenticity of the attestation result.

For performance, we have the following goals,

- *Scalability* - The protocol shall be scalable to large number of network nodes in terms of both computation and network bandwidth consumption.
- *Heterogeneity* - The protocol shall support devices with different radio contexts.
- *Energy Efficiency* - The protocol shall be energy efficient. Many of the network nodes in CRN are battery-powered. Energy remains one of the biggest challenges in mobile computing.
- *High Fidelity* - The protocol shall be able to provide system verifier not only the aggregated statistics, but also the exact list of violating devices as well as their radio contexts.

III. ROSTER Protocol

ROSTER is a network attestation protocol designed to provide network scale remote attestation on radio context in cognitive radio networks. The attestation protocol includes *key initialization* and *network attestation*. In key initialization, the secret keys are established for all the network entities, i.e., the provers - CR devices, base stations, SASs and verifier. These keys are used for the network attestation later. We assume the higher layer topology between base stations, SASs and verifier is fixed. However, CR may connect to different base stations. In network attestation, the verifier obtains a cryptographically verifiable measurement of the radio contexts for nodes in the network. The parameters of the protocol are defined in Table. I.

Participant IDs	
RA	Regulatory authority
Ver	Attestation verifier
SAS_q	SAS, either civilian or opsec-aware
B_i	Base station i in the network
d_j	CR Device j
Key materials	
k_j	Shared secret key between CR d_j and its base station
k_{B_i}	Shared secret key between B_i and its associated SAS
k_{S_q}	Shared secret key between SAS_q and verifier
$k_{B_i,V}$	Report key used to sign the attestation report to verifier by base station B_i
Device variables	
S_j	Software configuration of d_j
R_j	Radio configuration of d_j
L_j	Location measurement of d_j
M_j	Radio context measurement of d_j, i.e. $M_j = \{S_j, R_j, L_j\}$
Parameters	
τ	Attestation token
t_e	Expiration time for an attestation token
ctr	Monotonically increased counter value for an attestation token
σ	Signature on the token expiration time and counter value by RA
δ_q	MAC generated by SAS_q using k_{B_i}
mac_j	MAC generated by d_j using k_j
MAC_i	MAC generated by B_i using k_{B_i}
$CC = S\|R\|L\|I\|RC$	A 5-bit context check field indicating check results for software, radio, location, identity, and radio context
rep_{B_i}	Report to verifier by B_i
rep_{S_q}	Report to verifier by SAS_q

TABLE I. Parameter definition

A. Key Initialization

Three types of shared keys are generated during this step. First type is the shared key between CR and base station. Upon registering to the base station B_i, CR d_j negotiates a key k_j with B_i. This key is bootstrapped by the code inside the secure component of the device and protected by the hardware. The second type is the shared secret key k_{B_i} between the base station B_i and SAS_q. The third key is the shared secret key k_{S_q} between SAS_q and the cloud verifier Ver. The shared secret can be generated by either utilizing existing PKI [10],

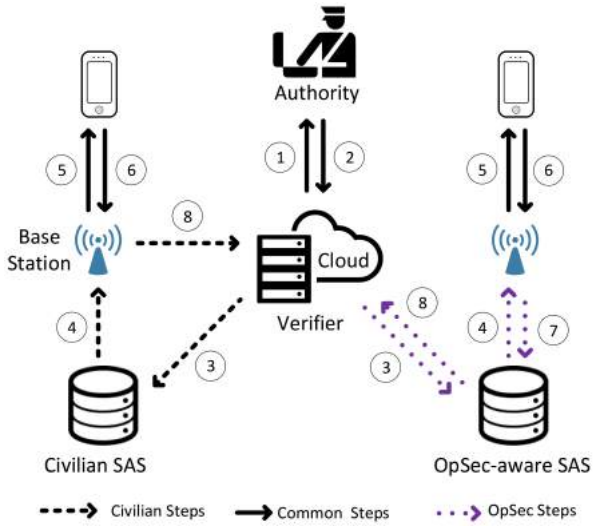

Fig. 2. ROSTER Protocol Overview

or adopting a key distribution mechanism [14]. The shared keys are used later for attestation. Formally,

$$\mathsf{keySetup}[(pk, sk) \text{ for } d_j, B_i, SAS_q, Ver] \rightarrow [d_j : k_j; \\ B_i : (k_j, k_{B_i}); SAS_q : (k_{B_i}, k_{S_q}); Ver : k_{S_q}].$$

B. Network Attestation

Fig. 2 shows the high level message exchange for radio context attestation in ROSTER. The online attestation can be logically divided into three phases. Steps ① to ④ represent the ***attestation request propagation*** in section III-B2, steps ⑤ and ⑥ are ***radio context measurement*** in section III-B3, and steps ⑦ and ⑧ are ***context auditing and aggregation*** in section III-B4.

1) Network Attestation Overview: Verifier in cloud first obtains the attestation token from the authority in steps ① and ②. Verifier then sends attestation request with the attesation token to SASs in step ③. However, due to sensitivity of the spectrum availability information, the rest of the protocol is different between opsec-aware SAS and civilian SAS. Civilian SAS attestation steps are shown on the left of Fig. 2 in black, while the opsec-aware SAS steps are on the right in purple. For civilian SAS, the spectrum availability is not sensitive, therefore it is forwarded to base station in step ④. After the radio context measurement in step ⑤ and step ⑥, base station will audit the result with the spectrum information, and forward the results back to verifier in step ⑧. For opsec-aware SAS, the spectrum availability cannot be distributed out to base station during step ④. Therefore radio contexts measured in steps ⑤ and ⑥ have to be forwarded back to SAS in step ⑦ for auditing, and the result is reported back to verifier by the SAS in step ⑧.

2) Phase One - Attestation Request Propagation: Steps ① to ④ capture the first phase of ROSTER. In steps ① and ②, an authorized verifier requests an attestation token from RA. The legitimate Ver should be given a valid token τ after mutual authentication with RA. The secret to initiate network attestation is separated from the verifier such that regulatory authority, such as FCC or DHS, can reap the benefit of elastic computing offered by public cloud. τ includes an expiration time t_e, a continually increased counter ctr and the signature $\sigma = \text{Sign}_{sk_{RA}}(t_e, ctr)$ from RA. The expiration time and counter will prevent adversary from reusing old attestation tokens, thus stopping potential DoS attack by adversary spamming attestation request on the network. In step ③, Ver sends the selected SASs the individual attestation requests, consisting of the token $\tau = \{t_e, ctr, \sigma\}$ and a nonce N_v.

In step ④, upon receiving the request, each SAS calls verifyRequest() function to verify the signature using RA's public key and check the included expiration time to ensure the freshness of this token. It also checks if the counter value in the token is greater than the stored value and accordingly update this value for the next attestation request. Failure of any of the above steps will result in protocol termination. In a conventional software-based only network attestation, it is often sufficient to propagate only the attestation token and a single configuration hash down the chain. However, due to the dynamic characteristic of radio context, it is not possible for the base station to know what type of configuration is legal without the full spectrum information, which only SAS possesses.

For civilian SAS, the spectrum information is not sensitive. Therefore, it can be forwarded to the base station B_i such that the measurements can be directly audited on the base station and sent back to the verifier. However, there is no pre-shared key between base station and the verifier. Furthermore, there are millions of base stations operated by different carriers [15], [16], key management between the verifier and individual base station is a challenge. To solve this problem, SAS generates the report key $k_{B_i,V} = prf_{k_{S_q}}(B_i)$ for attestation reporting from B_i to Ver, where prf is a pseudo-random function. In particular, the SAS invokes updRequest() function with input $k_{B_i,V}$ along with the related spectrum availability information inf in the attestation request. The SAS also computes $\delta_q = \text{MAC}_{k_{B_i}}(inf, k_{B_i,V})$ and adds it to the request. In the end, the corresponding base station acquires the report key $k_{B_i,V}$ and additional knowledge for verification in step ④. This key derivation and attestation request update process can be formally represented by

$$\mathsf{infUpd}[B_i : -; SAS_q : k_{S_q}, inf, \{\tau, N_v\}; Ver : k_{S_q}] \\ \rightarrow [B_i : \{\tau, N_v, k_{B_i,V}, inf, \delta_q\}; SAS_q : -; Ver : k_{B_i,V}].$$

On the other hand, for opsec-aware SAS, spectrum information is sensitive and can not be disclosed. In this case, the attestation request to base station in step ④ contains only the attestation token τ and N_v. Radio context audit will be performed by the SAS.

3) Phase Two - Radio Context Measurement: Steps ⑤ and ⑥ are the radio context measurement process in ROSTER. Upon receiving the attestation request, each base station will call verifyRequest() to perform the token validation as described for the SAS. If correctly verified, it will continue to perform the remote attestation on the connected CRs. However, different from the software remote attestation, ROSTER aims to measure the radio contexts of devices. The context is measured by the attestation routine inside the trusted container of the device. The results are then signed and delivered to the base station. More specifically, the base station forwards the

attestation request to the connected CRs. If the request tokens are successfully verified, CR d_j will then produce the radio context measurement M_j. The measurement is composed of three parts: software configuration S_j, radio configuration R_j and location L_j. Time is implicit at the execution time of the protocol. CR then generates the response $\{mac_j, M_j, d_j\}$, where $mac_j = \text{MAC}_{k_j}(d_j, M_j, N_v)$, using the shared key k_j with the base station. Upon receipt of the response, the base station can verify the measurement as follows. First, it verifies the received mac_j. If successfully verified, the response from the CR is authenticated, i.e. it indeed comes from the expected device. This is because the MAC key k_j is stored in the secure component of d_j and only in this secure environment that the MAC can be produced. More formally, the process can be represented by

$$\begin{aligned}&\mathsf{devResponse}[d_j : \tau, N_v; B_i : -; SAS_q : -; Ver : -]\\ &\quad \rightarrow [d_j : -; B_i : mac_j, M_j, d_j; SAS_q : -; Ver : -].\end{aligned}$$

4) Phase Three - Context Auditing and Aggregation: Steps ⑦ and ⑧ make up the context auditing and aggregation phase. Upon verification of the authenticity of the context measurement from CR, the base station will need to verify that the contexts are compliant. Depending on the sensitivity of the SAS serving the base station, there are two cases.

For base stations connected to civilian SAS, it will perform context auditing with the spectrum information and transmit the results directly back the verifier in step ⑧. To verify the compliance of a connected CR, the base station B_i first checks the location measurement. Taking advantage of the physical proximity between base station and the CR, the base station can estimate the relative distance and direction of a connected CR. Using its own location as a reference, the base station can calculate the approximate location of the CR. If the reported location measurement is different than what is measured by the base station, then the reported location from CR is inaccurate. The software configuration S_j can be verified by checking against a set of known device software configurations, if S_j is not on the list, then it is likely that the CR platform or application software is modified. For radio configuration, there is no known list of compliant configurations due to dynamic spectrum availability. With the related spectrum information inf received in the attestation request message, the base station B_i can determine if the R_j is compliant or not, based on inf, S_j, R_j. When the radio context of a CR is compliant, only the device ID d_j will be incorporated into the final report. Otherwise, the base station will also submit the measurement M_j and the context check field CC as indicated in Table. I. Specifically, any successful check will set the corresponding bit of this field. For example, $CC = 10111$ indicates that all the verification steps are successful except the radio configuration check. Thus the final attestation report in the case of one honest device d_1 and one malicious device d_2 will be generated as $rep_{B_i} = \{\text{MAC}_{k_{B_i,V}}(msg|N_v), msg\}$, where $msg = d_1|(d_2, M_2, CC_2)$. Then $\{rep_{B_i}, msg\}$ will be sent to the cloud verifier in step ⑧. Formally,

$$\begin{aligned}&\mathsf{attReport}_1[d_j : mac_j; B_i : inf, k_j, k_{B_i,V}; SAS_q : -; Ver : -]\\ &\quad \rightarrow [d_j : -; B_i : -; SAS_q : -; Ver : rep_{B_i}].\end{aligned}$$

For base stations connected to opsec-aware SAS, the base station is only able to verify the location measurement and identity of the responded device. B_i sets the corresponding bits in CC accordingly and produces its response similar to rep_{B_i}, except that the base station uses the shared secret key k_{B_i} instead of $k_{B_i,V}$ to calculate the message authentication code MAC_i. The report is then sent to the associated opsec-aware SAS for further auditing in step ⑦. After successfully verifying the received MAC from B_i, the SAS continues the check on software configuration and radio configuration of the radio context. It then sets the corresponding bits of CC if the device is believed to be non-compliant and sends the final attestation report rep_{S_q} to the cloud. rep_{S_q} is derived using secret key k_{S_q} similarly to rep_{B_i}. Note that the SAS can further aggregate the attestation results under its supervision instead of generating one report for each of its connected base station. The aggregated report is then sent to the verifier in step ⑧. Formally,

$$\begin{aligned}&\mathsf{attReport}_2[d_j : mac_j; B_i : k_j, k_{B_i}; SAS_q : k_{B_i}, k_{S_q};\\ &\quad Ver : -] \rightarrow [d_j : -; B_i : -; SAS_q : -; Ver : rep_{S_q}].\end{aligned}$$

Lastly, the verifier can check the authenticity of the received reports by examining the MACs. One of the key motivations to include individual attestation result in the report rep rather than a binary decision of whether the node is malicious or not [10], [11] is to enable risk mitigation. System triage is often a missing component in intrusion detection system, it is however one of most important steps in network management. If a device has malicious software, the verifier can send the report to device operator to push software updates to the device. If a device is found to be falsifying its location, the SAS may want to disable further spectrum allocation to the device until a corrective action is in place.

IV. Security Analysis

The security goal of ROSTER is to allow the verifier Ver, a regulatory entity, to measure and audit the radio contexts of connected CRs in CRN. We formalize the security goal as an experiment $\text{Exp}_{\mathcal{A}}$ between the adversary $\mathcal{A}$, the network $\mathcal{N}$ and the verifier Ver. $\mathcal{A}$ is able to compromise at least one CRN device and modify its measurement M. In addition, he can insert, delete, and modify the messages transmitted in $\mathcal{N}$. However, $\mathcal{A}$ cannot control base stations and SASs. The attack is allowed to proceed with a polynomial number of steps before Ver outputs a 1-bit result b. $b = 1$ indicates Ver accepts the attestation result; $b = 0$ otherwise. We define the security of a CRN attestation scheme as follows:

Definition 1. *A CRN attestation scheme is secure if for any probabilistic polynomial time (PPT) adversary $\mathcal{A}$, the probability that $\mathcal{A}$ succeeds in the experiment is negligible, i.e., $Pr[b = 1|Exp_{\mathcal{A}}(\lambda) = b] \leq\ negl(\lambda)$*

Theorem 1. *(Security of ROSTER) ROSTER is a secure CRN attestation protocol if the underlying signature and MAC schemes are unforgeable.*

The verifier Ver accepts the received reports rep_{B_i} and rep_{S_q} if it can successfully verify the message authentication codes $\text{MAC}_{k_{B_i,V}}(msg_{B_i}|N_v)$ and $\text{MAC}_{k_{S_q,V}}(msg_{S_q}|N_v)$. The messages in the reports include CRN device IDs, as well as the measurement M_j comprising software configuration

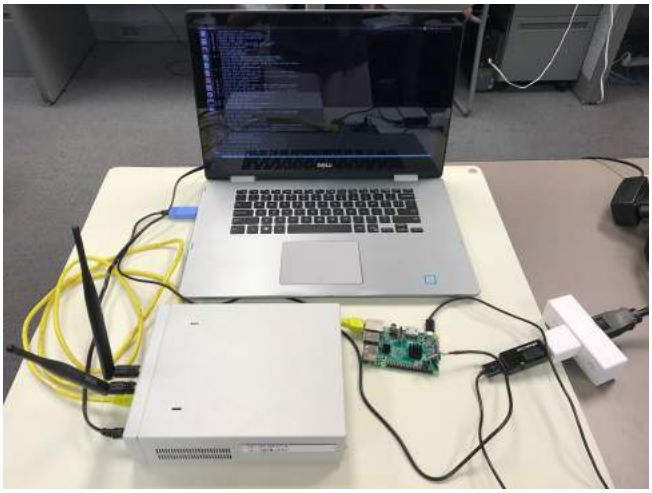

Fig. 3. ROSTER Prototype Hardware Platform

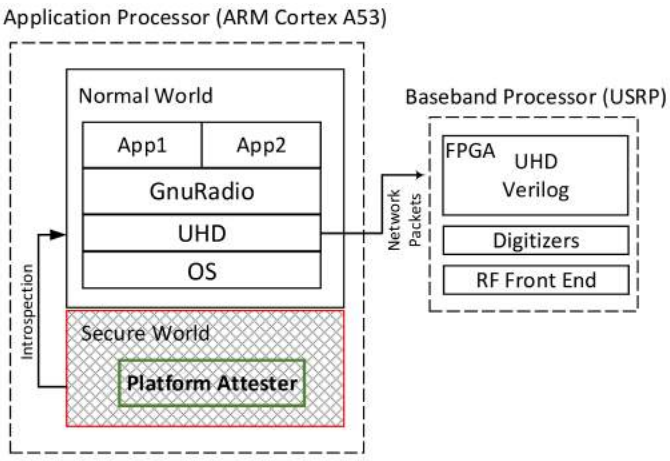

Fig. 4. ROSTER Prototype Software Architecture

S_j, radio configuration R_j and location measurement L_j, and the context check field CC_j if d_j is compromised. First, counterfeiting the attestation request is not viable. This is because $\mathcal{A}$ cannot access the secret keys in the experiment. Thus, the signature σ of RA and δ_q from SAS are unforgeable. We consider three cases where the adversary $\mathcal{A}$ may launch attacks for attestation report process.

The first is that $\mathcal{A}$ may compromise at least one device. Thus, it can modify the reported device ID, S, R or L. Since $\mathcal{A}$ cannot temper with the attestation code and access the secret key k_j in the secure component, the modified R' and S' will be different from R and S measured by secure hardware. Similarly, the modified L' is different from L measured by its associated base station. All these attacks need forge mac_j, which incurs a negligible probability due to the unforgeability of MAC. It also implies the probability of replacing ID is negligible.

In the second case, $\mathcal{A}$ is able to temper with the communication links in $\mathcal{N}$. Specifically, it can add, delete, and modify messages transmitted between d_j and B_i, B_i and SAS, B_i and Ver, SAS and Ver. We assume that the base station and SAS are both well protected and that the secure component cannot be compromised by $\mathcal{A}$. In the situation of compromising the link between d_j and B_i, adding and modifying the response from d_j is equivalent to the first case of compromising the device. Deleting the response will be eventually detected by the base station when the expected return time expires. Likewise, $\mathcal{A}$ will not successfully launch the attack either in cases of compromising links between B_i and SAS, B_i and Ver, SAS and Ver because the probability of successfully forging the underlying MACs is negligible in the security parameter.

$\mathcal{A}$ in the third case can compromise the device and links at the same time. This is a combination of case 1 and case 2. In light of the same reason, the adversary cannot make it in this situation either.

Note that the adversary can also launch replay attack in the above three cases. However, since the random nonce N_v issued by Ver is incorporated in the MAC of final report and the MAC is unforgeable, Ver will detect the misbehavior at the final verification phase.

V. ROSTER Implementation

The experimental hardware is shown in Fig. 3. We build a prototype of cognitive radio using Raspberry Pi 3 as the application processor and USRP platform as the baseband processor of the cognitive radio device. We picked Raspberry Pi as the application processor due to it is low unit-cost. In order to achieve high adoption rate, we believe our proposed system should not require high-end hardware. USRP has been one of the de facto experimental platform for cognitive radio research. For base station, SAS and verifier, we use t2.micro VM instances on Amazon Web Service (AWS). The software architecture is shown in Fig. 4. TrustZone [13] is used to build an isolated environment for the attestation software. TrustZone is an architectural security extension on ARM systems that consists of modifications to processor, memory and peripherals. It is widely available in most of the modern ARM system-on-chip application processors [13]. Our prototype runs the OP_TEE [17] secure kernel in the secure world, and Ubuntu 15.04 with 4.6.3 ARM 64 bit Linaro Linux kernel in normal world. The platform attestation code is implemented as an OP_TEE static trusted module with approximately 1000 software line of code (SLOC). We refactored the openSSL 1.0.1f library to extract only the cryptographic functions needed for our protocol implementation. The network communication is assisted by the normal world as proxy. Therefore the trusted computing base (TCB) of the device consists of the platform attestation software, the secure operating system from OP_TEE as well as the TrustZone-enabled ARM hardware. We also install GnuRadio in the normal world to communicate with the USRP hardware. For software configuration, we measure the SHA256 checksum of the code page of the Operating System kernel. For radio configuration, we measure the checksum for global variables as well as libUhd library code pages inside the communicating process, because libUhd is the device driver for USRP in the GnuRadio software stack.

It should be noted that our prototype implementation uses

HW	Function	Time(ms)	Energy(J)
Pi	SW Check (SHA256)	19.7	0.0065
Pi	Radio Check (SHA256)	77.09	0.039
Pi	HMAC	0.12	0.0005
Pi	RSA 2048 Sign	22.3	0.04
Pi	Pairing Sign (d224)	4.8	0.009
AWS	HMAC(SHA256)	0.024	—
AWS	RSA 2048 Sign	0.814	—
AWS	RSA 2048 Verify	0.035	—
AWS	Pairing (d224)	6.181	—

TABLE II. MICRO-BENCHMARKS ON CRYPTOGRAPHIC FUNCTIONS

Server	Client	B(Mb/s)	L(ms)	J(ms)	PL(%)
Campus	AWS(OR)	94.3	66.35	1.004	0
AWS(OR)	Campus	50.2	67.09	0.042	0
AWS(VA)	Campus	94.2	2.83	0.034	0
Mobile	Campus	21.4	148	8.715	0.11
Mobile	AWS(OR)	16.7	255	7.656	3.3
Mobile	AWS(VA)	16.9	125	7.928	3.3
AWS(VA)	AWS(OR)	96.6	83.3	0.062	0.0019
AWS(OR)	AWS(VA)	90.5	83.21	0.030	0.0019

TABLE III. NETWORK MEASUREMENT - BANDWIDTH(B), LATENCY(L), JITTER(J), PL(PACKET LOST)

checksum of the memory as measurement. While it is a common approach, malicious attackers can use advanced software attack techniques such as return-oriented programming [18] to compromise the process without modifying the memory pages of program code. Capturing the context of program execution remains an active area of research [19].

VI. PERFORMANCE EVALUATION

We evaluate the proposed attestation protocol at both micro-level and macro-level. At micro-level, we benchmark the prototypes. At the macro-level, we perform network simulation to study the effects of different settings on the protocol execution.

A. MicroLevel - Prototype Benchmarks

Device benchmark measures computation and energy cost for various functions in radio context attestation. Bandwidth and delay are measured in network benchmark to provide estimation for the simulations.

1) Device Benchmarks: Table II shows our measurement of each function in the attestation protocol on the prototype. The prototype is built with Raspberry Pi3 and Amazon AWS cloud t2.micro instance. Raspberry Pi3 measurements provide insights to the cost of the attestation on cognitive radio devices, while the measurements on Amazon cloud instances are good indication for the cost at the SAS and verifier. Since CR nodes are often battery-powered, power consumption is major concern for protocol design. Therefore, we measure not only the computation cost (in time) but also the energy spent on the attestation function. The separated measurement of time and energy is necessary since processor usage doesn't always indicate power usage [20]. The energy usage of the platform is measured with a USB power unit adopter as shown in Fig. 3. All the measurements shown in Table. II are an average of 10 experiments on 10000 repetitions of the same function.

Contrary to our expectation, the main power consumption is due to software introspection rather than the cryptographic operations. In our prototype, the software introspection process involves calculating the checksum of the memory pages of kernel code. This involves running SHA256 on approximately 1 MB of kernel memory. The radio check process involves calculating the SHA256 checksum of libUhd library of GnuRadio. The size of the library used in our prototype has a 3.5MB code page, this is because the version we used is compiled to support different hardware. Therefore it takes longer to calculate the checksum. In deployed system, the binary can be optimized to reduce the size. On the other hand, the main cryptographic function used in our protocol on the cloud is HMAC, which is very efficient and will not be a bottle neck.

Besides the primary function in ROSTER, several other cryptographic primitives are also measured to provide supplemental information on our design choice. For example, if public key cryptography, such as RSA, is used as the signature scheme for device level attestation, we will experience orders of magnitude extra energy consumption for message signing. However, the cost for cryptography remains manageable on the mobile device. Furthermore, we also measure the cost of pairing operations used in a closely related work [11]. While their solution is elegant for Swarm attestation. Direct adoption of same scheme to CRN would take hours to complete the signature check given the current mobile network size.

2) Network Benchmarks: The network benchmark is designed to provide a realistic expectation of the network condition for the proposed attestation protocol. Each link is measured in terms of bandwidth, latency, jitter and packet lost using iPerf benchmark [21]. These measurements are later fed into the network simulation.

We measure the network performance of mobile network using a laptop connected to Internet via USB tethering to an iPhone to perform the measurement. The mobile phone is subscribed to LTE service from the provider. We find that Amazon EC2 instances in Virginia are closer to the mobile phone switch than a computer node from campus network. Therefore, we use the measurement results from the mobile phone to EC2 instance in Virginia data center as connection to base station in the simulation. One measurement that is particularly interesting is the low uplink bandwidth from instances in Oregon computing center, upon further investigation, it is a common problem among users [22]. We further observe that the bandwidth between cloud instances is at the link capacity, therefore we assume full link capacity as advertised by Amazon in our network simulation.

B. MacroLevel - Network Simulation

To evaluate ROSTER on network level, we can no longer rely on individual benchmark from the prototype. Network

Simulator 3 [23] is used to simulate our system with different network configurations. Application level protocols are implemented on top of the NS3 programming framework, using bulk-send as a template. Cryptographic operations are modeled by manual delay by the amount measured in actual hardware listed in Table. II. The network links are modeled using channel statistics collected listed in Table. III. Due to limitations in NS3 and computing equipments, we were not able to fully simulate the entire network at the desired scale, subnet performances are measured separately and added as delay in upper networks.

1) Network Scale: To understand the scale of simulation necessary to test the feasibility of our proposed solution, we use the mobile ecosystem in United States as our baseline. The number of mobile subscribers in the US is 417.52 million [24]. The number of macro cells in the US is 320,000 [15], with each macro cells generally serving 1000 subscribers. The number of small cell is approximately 3.8 million [16]. 97.52 million subscribers are not supported by macro cells, therefore we estimated the average subscribers supported by small cell is 26. Despite our best effort to estimate the various setting in the network, we realize that both the complexity and scenarios will be different in real deployment. The evaluation is also available on our git hub page for further investigation in the community. The infection rate of mobile device over the first half of 2017 is estimated to be 1% [25].

2) Attestation Communication Cost: Bandwidth consumption is one of the major problems when attestation is performed over large number nodes in the network. Attestation report includes a list of IDs for all the devices attested to have a compliant configuration and a list of non-compliant devices with the corresponding radio context. Therefore, the bandwidth cost is directly related to the length of the ID as well as the size of radio context. The radio ID is stored as an 8 byte value, since the current widely used mobile equipment identifier (MEID) is 56 bits [26]. 64 bits is used to accommodate future growth in the number of mobile devices. The radio context contains the measure for software, radio as well the location. In our evaluation, the size of radio context M is 67 bytes. The software context is the hash of kernel code, which is 256 bits long for SHA256. Hash of the same length is also used to capture the configuration of the radio. Location is stored as GPS coordinate with two decimal precision, which is 16 bits. Therefore the radio context of a CR is 66 bytes in our prototype. The last byte is used to store the attestation check CC.

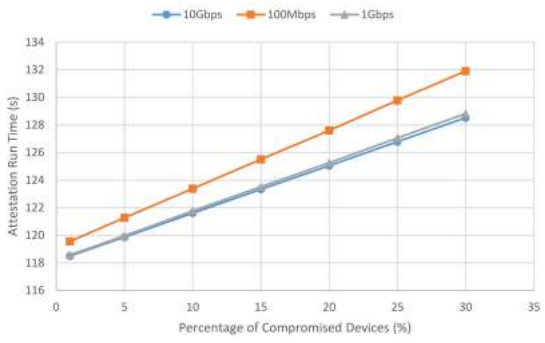

Fig. 5. Effect of Compromised Device on Attestation

3) Simulation Result: Fig. 5 shows the effect of compromised devices on the attestation run time. On the x-axis is the percentage of compromised CR, while the y-axis shows the normalized overhead of attestation run time. We use a simple scenario where there is only one opsec-aware SAS and one civilian SAS. All simulation parameters are hold constant except the percentage of the compromised devices. With the number of compromised devices increased, we expect to see the attestation taking longer, because it is now necessary to report the additional details on the compromised devices. We are also interested in the impact of bandwidth with increasing number of compromised devices. As can be observed in the figure, the overhead is independent of the bandwidth and grows linearly with the number of compromised CR.

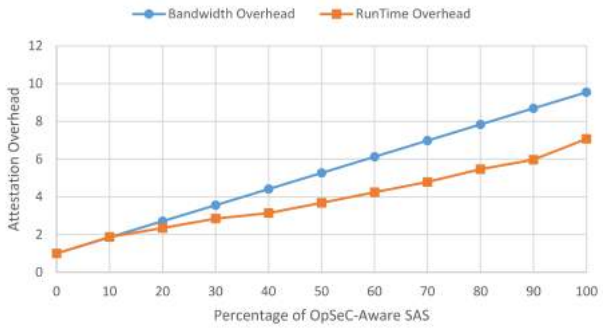

Fig. 6. Runtime Impact of opsec-aware SAS

Figure 6 shows the normalized overhead of using opsec-aware SAS. Opsec-aware SAS requires the full configuration of all devices during radio context auditing, therefore incurs significant bandwidth overhead. X-axis shows the percentage of opsec-aware SAS in the CRN, and the y-axis shows the normalized overhead. We investigated the overhead for both attestation run time and bandwidth. The maximum overhead for bandwidth is about 10x, while the maximum overhead for runtime is about 6.5x. The difference between runtime overhead and bandwidth overhead, we believe, is due to the basic connection establishment and maintenance.

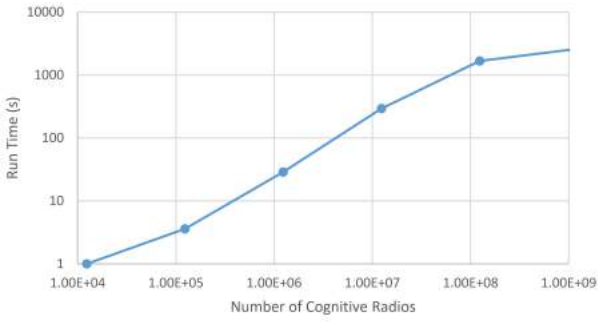

Fig. 7. Attestation Scalability

Figure 7 shows the our simulation study on the scalability of ROSTER. X-axis shows the number of CR in CRN, y-axis shows the normalized attestation run time in the simulation. We can see that ROSTER scales linearly to the number of SDR nodes supported.

VII. Related Work

While we are the first to coin the term Radio Context and to investigate remote attestation on the context, our research is closely related to CRN security and remote attestation.

Recognizing the importance of security of CR, there has been significant amount of research efforts in the general area of security in CRN [3], [27], [4], [6], [5], [28], [7], [8], [29], [30]. Providing authentication and non-repudiation in cognitive radio network has been one of the most studied areas in CRN security. Features are embedded into the waveforms to provide unique signatures for authentication. These features can either be intrinsic features [28] which originate from the physical property of the radio or extrinsic features which are injected artificially [6], [3], [5], [4]. Security and privacy of user location is also studied in [29], [30]. Lastly, there are also two closely related research efforts on securing the CR [7], [8]. In [7], a host-based anomaly detection system is proposed in application space to monitor cognitive radio applications on an artificial malware dataset. A secure reconfiguration software architecture based on virtualization is proposed in [8]. ROSTER employs hardware-enabled security containers, and the primary objective to provide network wide measurement and auditing rather than device level intrusion detection.

The second closely related area is remote software attestation, which is a well studied subject [19]. While individual device attestation is mature, network attestation remains an unexplored area. Two recent works [10], [11] in this area are closely related to ROSTER. SEDA [10] is symmetric key based swam attestation protocol, while SANA [11] is asymmetric key based. These two protocols are designed for ad-hoc network software attestation, while the network for ROSTER is infrastructure-based. Signature aggregation scheme [31] used in SANA is particularly effective in aggregating attestation report signatures. Unfortunately, the heterogeneity in radio context makes it prohibitive to apply pairing-based signature aggregation in ROSTER.

VIII. Conclusion

In this paper, we propose ROSTER, a radio context attestation protocol for CRN. We call the collective context of device software configuration, radio configuration, location and time the radio context. The ability to obtain cryptographically provable measurement of cognitive radio compliance is a fundamental capability in CRN security. ROSTER is the first to tackle this challenge. Our design on the attestation protocol takes in consideration of the unique network and security requirements in CRN. It is computation, energy and network bandwidth efficient. ROSTER is evaluated with both system prototype and network emulation to provide insights into strength and weakness of attestation protocol and offer direction for the future design of CRN attestation.

Acknowledgment

This work was supported in part by US National Science Foundation under grants CNS-1446478 and CNS-1443889.

References

[1] FCC, "FNPRM: Ammendment of the commissions rules with regard to commercial operations in the 3550-3650 mhz band," 2014.

[2] X. Jin, J. Sun, R. Zhang, Y. Zhang, and C. Zhang, "Specguard: Spectrum misuse detection in dynamic spectrum access systems," in *IEEE INFOCOM*, 2015.

[3] X. Jin, J. Sun, R. Zhang, and Y. Zhang, "Safedsa: Safeguard dynamic spectrum access against fake secondary users," in *ACM CCS*, 2015.

[4] V. Kumar, J.-M. Park, and K. Bian, "Blind transmitter authentication for spectrum security and enforcement," in *ACM CCS*, 2014.

[5] L. Yang et al., "Enforcing dynamic spectrum access with spectrum permits," in *ACM MobiHoc*, 2012.

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

[7] Y. Dou, K. C. Zeng, Y. Yang, and D. D. Yao, "Madecr: Correlation-based malware detection for cognitive radio," in *IEEE INFOCOM*, 2015.

[8] C. Li, A. Raghunathan, and N. K. Jha, "An architecture for secure software defined radio," in *DATE*, EDAA, 2009.

[9] A. Seshadri, A. Perrig, L. van Doorn, and P. Khosla, "Swatt: software-based attestation for embedded devices," in *IEEE S&P*, May 2004.

[10] N. Asokan, F. Brasser, A. Ibrahim, A.-R. Sadeghi, M. Schunter, G. Tsudik, and C. Wachsmann, "Seda: Scalable embedded device attestation," in *ACM CCS*, 2015.

[11] M. Ambrosin, M. Conti, A. Ibrahim, G. Neven, A.-R. Sadeghi, and M. Schunter, "Sana: Secure and scalable aggregate network attestation," in *ACM CCS*, 2016.

[12] P. Kocher, R. Lee, G. McGraw, A. Raghunathan, and S. Moderator-Ravi, "Security as a new dimension in embedded system design," in *ACM DAC*, 2004.

[13] "ARM Security Technology, Building a Secure System using TrustZone Technology," apr 2009.

[14] S. A. Camtepe and B. Yener, "Key distribution mechanisms for wireless sensor networks: a survey," *RPI Technical Report*, 2005.

[15] "Steel in the air." https://goo.gl/Jvo98U.

[16] "Small cell forum - market status report." http://scf.io/en/white_papers/Market_status_report_June_2017_Special_edition.php.

[17] "Open portable trusted execution environment." https://goo.gl/oyJsfv.

[18] H. Shacham, "The geometry of innocent flesh on the bone: Return-into-libc without function calls (on the x86)," in *ACM CCS*, 2007.

[19] R. V. Steiner and E. Lupu, "Attestation in wireless sensor networks: A survey," *ACM Computing Surveys (CSUR)*, vol. 49, no. 3, p. 51, 2016.

[20] Z. Xu, H. Wang, Z. Xu, and X. Wang, "Power attack: An increasing threat to data centers," 2014.

[21] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs, "Iperf: The tcp/udp bandwidth measurement tool," 2005.

[22] A. Zhitnitsky, "Amazon ec2 2015 benchmark: Testing speeds between aws ec2 and s3 regions." https://goo.gl/QwXFVb.

[23] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," *Modeling and tools for network simulation*, pp. 15–34, 2010.

[24] "Number of subscribers to wireless carriers in the u.s. from 1st quarter 2013 to 1st quarter 2017, by carrier (in millions)." https://www.statista.com/statistics/283507/ubscribers-to-top-wireless-carriers-in-the-us/.

[25] "Malware infection rate of smartphones is soaring android devices often the target." https://goo.gl/FESnnT.

[26] "3g mobile equipment identifier (meid)." https://www.3gpp2.org/Public_html/Specs/S.R0048-A_v4.0_050630.pdf, 2005.

[27] L. Xiao et al, "Using the physical layer for wireless authentication in time-variant channels," *IEEE TWC*, 2008.

[28] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *ACM MobiCom*, 2008.

[29] Z. Gao, H. Zhu, Y. Liu, M. Li, and Z. Cao, "Location privacy in database-driven cognitive radio networks: Attacks and countermeasures," in *IEEE INFOCOM*, 2013.

[30] K. Zeng, S. K. Ramesh, and Y. Yang, "Location spoofing attack and its countermeasures in database-driven cognitive radio networks," in *IEEE CNS*, 2014.

[31] D. Boneh, C. Gentry, B. Lynn, H. Shacham, *et al.*, "A survey of two signature aggregation techniques," *RSA cryptobytes*, 2003.