Volume 5, Issue 6
December 2016

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

Asynchronous Authentication of Helper Nodes in Cognitive Radio Networks with Stationary Incumbent Users

[1]Maged H. Ibrahim, [2]Fatty M. Salem

[1,2]Department of Electronics, Communications and Computers Engineering,

Faculty of Engineering, Helwan University

1, Sherif st., Helwan, Cairo, Egypt

E-mail: [1]mhii72@hotmail.com, [2]fatty4com@hotmail.com

ABSTRACT

In this paper, we focus on cognitive radio networks in which, the incumbent users are stationary and the authenticated reporting of incumbent users' spectrum information relies on sensor helper nodes, deployed (as a bridge) in close proximity to secondary users. We mainly address the problem of replay and wormhole attacks, which represent a challenging emulative attack in authenticating incumbent users to secondary users through helper nodes. Such attacks result from the possibility of partitioning helper nodes - which are distributed over a large geographical area -into isolated groups and relaying messages among the isolated groups via fast links, making use of the fact that synchronization over large geographic areas becomes loose. We introduce an authentication protocol phase between helper nodes and secondary users that when integrated with existing trust evaluation algorithms against Byzantine failure provides an efficient protocol to deliver spectrum information to secondary users in a correct and secure way. In our proposed protocol, the deployed helper nodes operate in a completely asynchronous fashion while the secondary users are not required to run expensive location verification tests to detect wormhole attacks.

Keywords: Cognitive Radio, Spectrum sensing, Authentication, Byzantine failure, Link signatures, Trust evaluation, Digital signatures.

1 INTRODUCTION

Cognitive radio (CR) networks allow unlicensed secondary users within the range of coverage of incumbent (licensed primary) users to utilize their licensed spectrum bands while causing no interference to incumbent communications [1, 2, 3, 4, 5]. The Federal Communications Commission (FCC) mandates that secondary users are allowed to access licensed bands as long as they do not interfere with the transmissions of incumbent users [6]. "Spectrum sensing" is one of the essential mechanisms and its operational aspects are being investigated actively [2, 4, 5, 7, 8, 9, 10]. By spectrum sensing, it is meant the process that allows secondary users to sense the spectrum white spaces (fallow bands) of incumbent users. However, existing spectrum sensing advantages increase the urge of adversaries to emulate incumbent users to mimic the characteristics of transmissions in order to reduce the bandwidth availability for honest secondary users aiming - for example - to preserve more bandwidth for her own use or to deny service to secondary users in the area. This raised "authentication" as an important security service that must be realized between the incumbent users and secondary users. Secondary users must ensure that the spectrum information (information about fallow bands) they receive are correct and originated from true incumbent users.

However, the technical challenge in applying any technique for CR signal authentication is that, according to the FCC specifications, no modifications are allowed on the incumbent users' side and their equipments. Also, the incumbent user’s surroundings (in the range of several meters) must be physically secure [5]. Following FCC recommendations, direct cryptographic solutions to the problem have become impossible.

The problem of authenticating incumbent users has recently received attention [11, 12, 13, 14, 15]. Several non-cryptographic solutions have been proposed. Link (or RF) signatures have been introduced as a non-cryptographic authentication technique used by secondary users, aiming to authenticate incumbent users based on the fingerprint of the communication link connecting them [16, 17]. However, link signatures alone are inaccurate due to the mobility of the secondary users. Moreover, it is expensive for secondary users to keep updating their information about the changes in link characteristics.

1.1 Cognitive radio with helper nodes:

A recent more promising solution [12, 13] is to deploy intermediate nodes, so called "helper nodes" in the area of coverage to help secondary users to authenticate to incumbent users. Helper nodes are stationary, and hence, link signatures are employed more efficiently and accurately to authenticate incumbent users to deliver spectrum information securely to helper nodes. Then, it is the role of helper nodes to deliver this information to secondary users in the area of coverage, also in an authenticated and correct way.

In our proposed protocol, we will employ link signatures as proposed in [16, 12, 13, 17] and we will follow the network topology in [13] as it is more efficient since it assumes that the helper nodes are deployed in the geographical area where secondary users exist and hence, they are always in close proximity to the secondary users requiring secondary users to communicate with low power; unlike the topology of [12] which assumes that helper nodes are deployed in close proximity to incumbent users. In this paper we focus on improving the authentication phase between helper nodes and secondary users to deliver authenticated spectrum information. We then apply existing efficient trust evaluation algorithms to efficiently withstand Byzantine failure in a non-interactive way.

1.2 Authentication service and Byzantine failure:

CR networks with helper nodes fall in the category of collaborative spectrum sensing networks, since a secondary user receives multiple reports about the fallow bands from many nearby nodes and hence, the network is vulnerable to Byzantine failure. In Byzantine failure, an adversary corrupting and/or emulating a helper node may submit false reports to disrupt the sensing process. There are well known cryptographic primitives in the area of secure multiparty computations and threshold cryptography [18, 19, 20 21, 22, 23, 24] that stand against Byzantine failure and achieve a Byzantine agreement, but unfortunately such protocols fail in cognitive radio due to several reasons: It requires extensive interaction among the parties (helper nodes in our case) which is undesired. Moreover, the computation complexity in secure multiparty computations prevents fast (real-time) delivery of the spectrum information reports. Another limitation is that, to apply secure multiparty computations, helper nodes must agree on a common report to deliver to secondary users in an authenticated way which is not the case in a cognitive radio network where each helper node has its own report to deliver.

In recent contributions concerning collaborative spectrum sensing, a secondary user must be able to weight the received reports based on the level of trust assigned to each node, this level is updated after every detection process according to a specific algorithm. The Byzantine failure problem in CR networks (and in wireless sensor networks in general) has been studied in the literature and efficient algorithms have been proposed, see for example [25, 26, 27]. Here, we must distinguish between two different (yet integrated) services; the authentication service and the Byzantine failure service. The reader may argue, why the authentication service is needed since the Byzantine failure algorithms are able to filter out false reports submitted by malicious adversaries? To answer this important question, we must distinguish between two different types of an adversary: an emulative adversary and a corruptive adversary. An emulative adversary is not able to corrupt an honest node, yet she can submit reports to the secondary user (in the absence of the authentication service) to disrupt the spectrum sensing process. On the other hand, a corruptive adversary is able to corrupt an honest node and completely controls it by sending authentic information (using the helper node's secret parameters) to the secondary users.

The task of the authentication service is to prevent an adversary from emulating an honest node since in this case, information sent to secondary users must pass the authentication verification algorithm, hence, the authentication service reduces the burden and improves efficiency and performance of the trust evaluation algorithms against Byzantine failure. Yet, still the adversary with enough power is able to corrupt and to fully masquerade honest nodes and hence, there is still a possibility that received authenticated reports are sent by corrupted nodes. The authentication service limits the task of the trust evaluation algorithms to withstand only corrupted nodes in a non-interactive fashion. Therefore, the task of our proposed authentication protocol is to prevent an adversary from emulating an honest node (specially through wormholes and replay attacks) then, after all received reports are verified as authentic, we recall efficient algorithm against Byzantine failure from [25] to filter out false (authentic-but-malicious) reports received from corrupted nodes. In the absence of node authentication an emulative adversary may easily deceive the weighted trust evaluation algorithms by emulating as many helper nodes as she can. As a result of the above discussion, the protocol of authenticating helper nodes to secondary users must be performed in two phases:

- Emulation prevention phase: Ensures that the reports are received from legitimate authenticated nodes, not from emulated ones. This phase is the main contribution of our paper.

- Trust evaluation phase: Filters out false reports received from authenticated but corrupted nodes. We recall an efficient algorithm from [25] to achieve this task.

1.3 Trust evaluation algorithms:

For clarity we give a brief description of the trust evaluation algorithms introduced in [25, 26]. In [26], the solution to the Byzantine problem in sensor networks is to model the network into a weight-based network, adapted in the architecture between a group of sensor nodes (SNs) and their forwarding node (FN) or aggregator. A weight W is assigned to each sensor node. The FN collects all information provided by SNs and calculates an aggregation result using the weight assigned to each SN: $E = \sum_{n=1}^{N}(W_n \times U_n)$, where E is the aggregation result, W_n is the weight ranging from 0 to 1 and Un is the sensor nodes output. In practice, the output information Un may be false or true information or continues numbers depending on the application where the sensor network is used. Let m be the total number of nodes in a cluster and s be the number of reporting nodes.

For a weighted penalty ratio *θ*, the SN's weight is updated as follows: $W_n = W_n - \theta \times r_n$ if $(U_n \neq E)$ and W_n is unchanged else wise, where *r_n= m/s*. To keep W_n in the range from 0 to 1, a simple normalization function is applied, $W_n = W_n / \max(W_1, \cdots, W_n)$.

A more efficient algorithm against Byzantine failure in cognitive radio with collaborative sensing is given in [25]. In [25], the computation of the suspicious level $\pi_n(t)$ provides a foundation for dealing with malicious nodes. The suspicious level is converted into trust level $\phi_n(t) = 1 - \pi_n(t)$. Trust value alone is not sufficient for determining whether a certain node report is reliable or not. In fact, the trust values become unstable when there is no enough observation or there is no malicious nodes. To solve this problem, consistency value of node *n* is defined as $\psi_n(t) = \sum_{\tau=1}^{t} (\phi_n(t) - \mu_n(t))^2$ where $(\sum_{\tau=1}^{t} \phi_n(t))/t$, for t < L and $\psi_n(t) = \sum_{\tau=t-L+1}^{t} (\phi_n(t) - \mu_n(t))^2$ where $(\sum_{\tau=t-L+1}^{t} \phi_n(t))/L$, for $t \geq L$, where L is the size of the window in which the variation of recent trust values is compared with overall trust value variation. Finally, the OR rule is applied after removing false reports according to the above procedure.

Paper organization. This paper is organized in seven sections as follows: In section 2, we give the related work achieved in this area of study. In section 3 we state our motivations and list the contributions of the paper. The assumptions and models are given in section 4. Our proposed protocol is described in section 5. The protocol security is analyzed in section 6. The simulation and results are shown in section 7 and finally, the conclusions are given in section 8.

2 RELATED WORK

Chen et al. [11] proposed an authentication method based on a network of monitoring nodes which verify the origin of incumbent user signals using received signal strength (RSS) measurements. If the estimated location of an incumbent user deviates from its known (and trained) location by a threshold, the signal is assumed to be emulated. However, location distinction methods based on RSS can be circumvented if the adversary employs antenna arrays [16].

Liu et al. [12] were the first to propose incumbent user's authentication system assisted by helper nodes deployed in close proximity to the incumbent users. The authors employed a combination of cryptographic and RF signatures to authenticate incumbent user's activity. In [12], helpers are physically bound to incumbent users which may be TV towers with thousands of watts of transmission power, covering an area of tens of square miles [3, 5]. Bi-directional communication between the helpers and the secondary users requires both types of devices to have communication ranges similar to that of TV towers. Their system is almost impractical for mobile devices with limited resources.

In the recent work of [13] helper nodes need only to be deployed within the area of the secondary users, independent of the location of the incumbent users. Hence, the secondary users communicate with low-power, which is the network configuration we assume in this paper. Moreover, in [12], a training phase is required before a secondary user can robustly sense incumbent users' activity. This phase must be repeated with every location change of the secondary users which is computationally inefficient and impractical. In [13], the problem is solved since the training phase is limited between the incumbent users and the helpers which are both stationary. Therefore, it does not need to be repeated due to secondary users' mobility, it is done only during initial deployment of helper nodes. However, since in [13] the helper nodes are deployed in close proximity to secondary users independently from incumbent users, the problem of isolation of helper node groups arise, which raises the vulnerability to wormhole attacks. This problem is solved in [13] using "helper node location resolution" algorithms which are computationally expensive and inaccurate again due to the mobility of secondary users.

Another major problem in [13] is that helper nodes over the whole network must be synchronized in order to withstand message replay attacks within the same area and message replay among isolated areas through wormholes. Since the distribution of the nodes depends on the secondary users' locations and that the geographical area is large, synchronization is hard to realize in practice and hence, given that the network will be at most loosely synchronized, replay and wormhole attacks come to play.

3 MOTIVATIONS & CONTRIBUTIONS

Motivations. The work in this paper is motivated by the observation that, although the recent work in [12, 13] introduced the idea of link signatures that give satisfactory and accurate results in the protocol phase of authenticating incumbent users to helper nodes, the protocol phase of authenticating helper nodes to secondary users is still inefficient. It requires all helper nodes in the network to be synchronized and since a cognitive radio network is usually large in practice (covering hundreds of miles), the nodes are loosely synchronized, giving the chance for successful wormhole attacks between two isolated networks. To detect wormhole attacks, each time spectrum information is received, the secondary users must run a "helper location resolution" algorithm to determine if two helper nodes are far apart [10, 13]. Due to the mobility of the secondary users, such algorithms (in addition to its complexity) gives inaccurate results and hence are not proven secure. Another way to track location of the helper nodes is through GPS location tracking, yet, such solution requires the secondary user to communicate with GPS frequently and hence, it is also an undesired solution. Contributions. We introduce an efficient protocol for authenticating incumbent users to secondary users in cognitive radio networks that rely on deploying helper nodes in the area of existence of secondary users. More precisely, we propose a new emulation prevention protocol phase that run between helper nodes and secondary users to ensure authenticated spectrum information delivery to secondary users. We introduce a new idea to efficiently realize this protocol phase in two rounds of communication. Then, after all reports are collected by a secondary user from the helper nodes in the area, we allow the secondary user to run a weighted trust evaluation algorithm to filter out false reports received from corrupted nodes. As a result, our protocol enjoys the following properties over the previous protocol proposed in [13]:

- It eliminates the synchronization requirement among all helper nodes in the network. The helper nodes operate in an asynchronous fashion and independent of each other.

- It is more efficient in resisting the challenging wormhole attacks without the need to perform any expensive and inaccurate "helper location resolution" algorithms by the secondary users. Such algorithms are complex and gives unsatisfactory results due to the mobility of the secondary user.

- Although our solution is almost purely cryptographic, it does not require the secondary users to store and manage any secret parameters.

- The secondary user initially stores only one public parameter (Deployment authority's public key) and hence the storage requirements is efficient. Computationally, the secondary user only performs several digital signature verifications which are of low complexity.

The basic idea of our protocol is described in section 5.1.

4 ASSUMPTIONS AND MODEL

In this section we clearly introduce our model, assumption, cryptographic tools and the efficiency considerations in our design.

4.1 Communication model

The entities playing roles in our proposed protocol scenario are:

- Deployment authority (DA): Responsible for deploying the helper nodes and loading them with necessary secret/public parameters and certificates. This is done during deployment phase.

- The incumbent user (IU): Representing the original legacy system, licensed to use a fixed spectrum bandwidth, which is divided into a set $B = \{1, 2, \cdots, \mathrm{B}\}$ of B orthogonal frequency bands, referred to as channels. The IU's are assumed to be stationary (e.g., TV or cellular towers) with a relatively large area of coverage.

- The secondary user (SU): Which is allowed to opportunistically use the set of channels if they do not cause interference on IU communications (fallow channels). The secondary users are assumed to be mobile and hence, limited in resources (e.g. batteries, storage capacity, computational capabilities, etc.)

- The helper node (HN): We assume that the number of helpers are enough to cover the geographical area where the SU's are located. The SU's are always able to connect to several HN's within the coverage area.

Figure 1, shows an example of a CR network configuration with helper nodes. The spectrum is the set of channels $B = \{1, 2, \cdots, 10\}$, the fallow spectrum channels are the set {3, 5, 9, 10}.

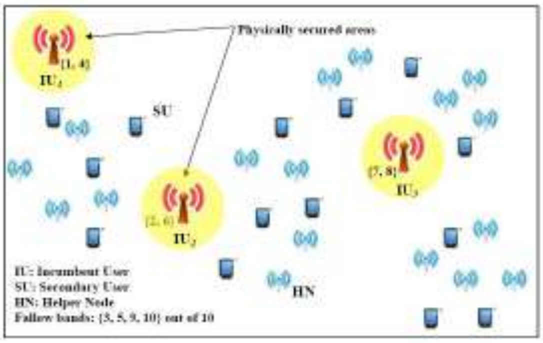

Figure 1. CR network with helper nodes

4.2 Adversary model and attacks

A secure stationary incumbent user cognitive radio authentication protocol must withstand the following types of possible attacks:

4.2.1 IU emulation attack

In a IU emulation attack, the adversary can try to impersonate the features of a IU signal on the idle portion of the spectrum. This can be achieved by mimicking features of IU transmissions such as power, modulation type, synchronization sequences etc., or by recording and replaying IU transmissions [4]. In this attack, the adversary must convince helpers that the emulated signal originates from an authentic IU.

As mandated by the FCC [6], IU's are assumed to be physically secured by restricting a forbidden area of several meters in radius around the IU's, hence, while the adversary can be present at any location within the coverage area and is equipped with software defined radio, she cannot emulate a transceiver in close proximity to an IU. Since an HN uses link (RF) signatures [11, 12, 13] to authenticate an IU, it is easy for an HN to robustly identify the IU's in its area, since we assume that all HN's are stationary, training the HN's for accurate identification is performed only once during deployment unless there are major changes in the environment which is unlikely to happen frequently. Therefore, the success of IU emulation is almost impossible.

4.2.2 HN attacks and assumptions

Since helper nodes are responsible for providing correct spectral information to the secondary users, we expect the urge of the adversary to either emulate (HN-emulation) or corrupt (HN-corruption) these nodes to disrupt the system and distribute false information of her choice aiming to limit the availability of the fallow spectrum to the SU's, or to destroy HN's to prevent SU's from receiving spectral information.

- HN emulation: An emulative adversary may attempt any of the following attacks:
    - Helper's message fabrication attack: The adversary may attempt to impersonate a helper in order to provide false occupancy information to the SU by fabricating false messages.
    - Helper's message replay attack: Without the opportunity of fabricating authentic messages, the adversary may choose to replay old ones, already broadcasted by the helpers to the SU's within range to falsify spectrum information.
    - Wormhole attack: The most challenging attack faced by cognitive radio networks and also ad/hoc networks. The adversary may replay spectrum authentication messages via a wormhole tunnel between two (or more) parts of the network [28]. The adversary deploys a fast link (wired or long-range wireless) between two separate (isolated) areas of the network. She then records broadcasted information from helpers on one end, transmits it via the wormhole tunnel to the other end and replays it to the existing SU's. The wormhole attack is shown in Figure 2, notice that in the previous protocol of [13] a single instant of successfully relayed false spectrum from area A has a dramatic impact on all secondary users in area B.

- HN corruption: The adversary completely controls a deployed honest node and uses the corrupted node to send false spectrum information to the secondary users. The message in this case is authentic-but-malicious. Authentic in the sense that it will correctly pass the authentication verification test as a cryptographically valid message and malicious in the sense that it contains false information (fabricated by the adversary) about the spectrum fallow bands.

Our proposed authentication protocol is the protocol phase that faces an adversary that has emulative capabilities against an HN. This type of an adversary is not able to reach any stored secret information in any deployed HN (in cryptographic terminology, this type of an adversary is spoken of as "non-corruptive adversary"). To face corruptive adversaries - after verifying authentication - the secondary user runs (as a second phase) an efficient trust evaluation algorithm against Byzantine failure from [25]. Withstanding destructive capabilities is achieved by deploying enough number (beyond the adversary's destructive capabilities) of HN's within the area of coverage.

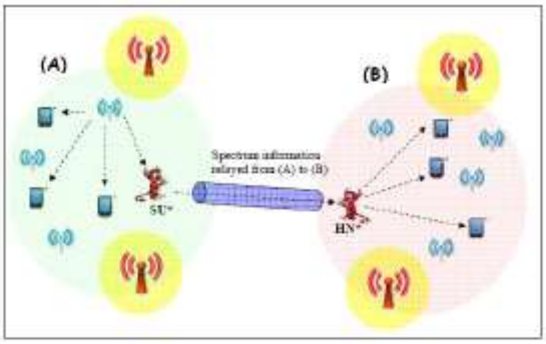

Figure 2. The wormhole attack and its impact in a synchronized network

Unlike the protocol in [13] that assumes all nodes are synchronized, our proposed protocol requires no synchronization among the HN's, HN's operate in an asynchronous fashion. Yet, our protocol is able to efficiently withstand replay and wormhole attacks without the need to run any computationally expensive "location resolution" algorithms by the SU's.

4.3 Efficiency considerations

Helper nodes interaction. In our protocol, the HN's are not required to interact together nor to be synchronized, HN's are required to interact with the IU to verify link signatures and with the SU's to send spectral information reports.

Secondary user efficiency. In our proposed protocols, the SU's never interact together, nor with the IU, they only interact with the HN's in close proximity and hence low level of transmission power is used. In the design of our protocol we keep in mind that the secondary users are mobile and hence, they are limited in resources. The following efficiency assumptions must be taken into consideration to avoid depleting the resources available to secondary users:

- The communication between an SU and an HN must be reduced to the minimum in the sense that, SU's must not be incorporated in many rounds of communications. In our protocol, it is required two rounds of communications between HN and SU to securely deliver spectrum information since, the SU is the one that initiates communication by polling the HN's in its neighborhood.

- No secret parameters is to be stored in an SU's device. The SU must not be incorporated in managing and protecting secret information. However, as in our proposed protocol, the SU may be able to generate a random nonce to challenge the HN's.

- The amount of public parameters (e.g. verification keys) that are required to be stored in an SU device must be reduced to the minimum since storage capacity of the SU's is limited. In our protocol, an SU needs only to store one public (verification) key and several digital signatures.
- The transmission power must be minimized to avoid batteries depletion. SU's communicates only with the nearest set of nodes.

4.4 Cryptographic tools

The readers who are familiar with digital signatures and public-key infrastructures may skip this subsection. We give a brief review of digital signatures as the cryptographic tool we will use in our protocol. It is required an unforgeable digital signature scheme resistive to chosen message attacks. Any standard digital signature algorithm (e.g. RSA, DSS, EC-signatures) will do the job [29. 30, 31, 32]. The choice is usually according to the security policies and regulations of the area.

4.4.1 Digital Signatures

A digital signature scheme consists mainly of three algorithms: the key generation algorithm which is usually performed by the certification authority given pre-established global parameters (if any), the signing algorithm *(sign(sk, M))* performed by the signer (holder of the secret key (*sk*)) and the verification algorithm (Verify(*pk, M, s*) performed by the verifier (holder of the corresponding public key (*pk*)), where *M* is the message to be signed. We choose the DSS and RSA signatures for our concrete (textbook) description of digital signatures.

Key generation: A DSS key is composed of a public key *pk* = *y* and a secret key *sk* = *x*, where:

- *x* is the secret key of the signer, a random 160 bit number less than *q*.
- $y = g^x \ mod\ p$ is the public verification key.

Signing algorithm: *sign(sk, M)*: Let m be a hash of the message M to be signed. The signer picks a random number *k* such that *1 < k < q*, calculates k^{-1} *mod q*, and sets $r = (g^{k^{-1}} \bmod p) \bmod q$ and *s = k(m + xr)mod q*. The pair (*r*; *s*) is the signature on M.

Verification algorithm: *Verify(pk, M, (r, s))*: A signature (r; s) of a message *M* can be publicly verified by checking that $r = (g^{ms^{-1}} y^{rs^{-1}} \bmod p) \bmod q$ where s^{-1} is computed modulo *q*.

An RSA digital signature scheme is as follows:

Key generation: An RSA is composed of a public key *pk = (e, N)* and a secret key *sk = (d, N)*, generated as follows:

- Pick two secret primes p and q each in the order of 512 bits and compute the RSA public modulus *N* = *pq* over the integers that is *N* is 1024 bits.
- Compute the RSA secret totient $\phi = (p-1)(q-1)$ over the integers.
- Pick $e \in_R Z_\phi^*$ that is $\gcd(e, \phi) = 1$.
- Compute $d = e^{-1} \bmod \phi$, using extended euclidian algorithm.

Signing algorithm: *sign(sk, M):* Let m be a hash of the message M to be signed. The signer computes $s = m^d \bmod N$, as the signature on M.

Verification algorithm, *Verify(pk, M, s):* A signature (*s*) of a message M can be publicly verified by checking that $s^e \bmod N = m$.

4.4.2 Certification authorities

The DA in our protocol plays a role similar to that of a root certificate authority (RCA) in a public key infrastructure (PKI). Briefly, the RCA is responsible for the generation of the secret/public key pairs *(sk_U, pk_U)* for each user. She then generates a certificate for the user's public key by signing the public key of the user using her own secret key sk_{CA} and generates the certificate:

$$cert_U = \{pk_U, sign(sk_{CA}, pk_U)\}$$

The above formula is actually a textbook formula that contains essential parameters, practical certificates include more parameters such as, user identity, time stamps, public-key algorithm in use, version, etc. What we want to emphasize for the purpose of our protocol is that, a signed message is sent from the signer to the verifier side by side with the certificate, any verifier that wants to verify a signature extracts the public key of the signer from the certificate he receives using the RCA public key pk_{CA}. This ensures to the verifier that the public key is consistent with the claimed identity.

5 OUR PROTOCOL

First we give a brief description of our idea, then we give the complete description of our protocol.

5.1 The basic idea

Our strategy in the authentication between SU's and HN's are different from previous protocols. In previous protocols, the HN's periodically broadcasts spectral information to all SU's in the area and hence, synchronization of common sequence numbers among all nodes in the network becomes a must to withstand replay attacks. Also, since the broadcast is a one round protocol, the wormhole attack becomes really fast and practical for the adversary to relay messages from one area to another taking advantage of the fact that the synchronization of the network on the long range becomes loose.

In our strategy, we do not let the HN's to periodically broadcast spectrum information, yet, only periodic updates of the spectrum information are allowed between HN's and IU's. In our protocol, an SU first poll s the HN's in the neighborhood by broadcasting a random nonce. This nonce will have two major benefits: First, it challenges the adversary to reply with a consistent digital signature including the random nonce as part of the signature. Second, it delays the wormhole attack to take much longer time (two rounds of communications) than the response from an honest node (one round). Broadcasting a polling random nonce by the SU is regarded as if the SU has started a race, the closer the node, the faster is its response. The node or (several nodes) that respond first win the race.

5.2 Protocol description

The protocol runs in three phases: DA-HN Deployment phase between the deployment authority and the helper nodes, IU-HN authentication phase between the incumbent user and the helper nodes, and HN-SU authentication phase between the helper node and the secondary user, which is the final (operational) and the most challenging phase representing our main contribution in this paper.

Now we are ready for the full description of our protocol (Figure 3), the protocol is as follows:

Deployment phase: In this phase, the deployment authority (DA) plays a role similar to that of a root certification authority of a PKI. She operates as follows:

- Sets up the digital signature global/public parameters (if any).

- Runs the key generation algorithm to generate her own secret/public key pair *(sk_{DA}, pk_{DA})*.

- For each helper node within the coverage area under the authority of DA, the DA runs the key generation algorithm to generate a secret/public key pair *(sk_{HA}, pk_{HA})* side by side with the *pk_{HA}'s* certificate *$cert_{HN}$*. where *$cert_{HN}$* is simply the signature of DA on *pk_{HN}* using her secret key sk_{DA},

  $Cert_{HN}$={pk_{HN} , sign(sk_{DA} ,pk_{HN})}

- On each deployed HN, the DA installs the tuple of secret and public parameters,

  T_{HN}= (sk_{HN}, pk_{HN}, $cert_{HN}$)

- The DA publishes her own certification public key pk_{DA} to all SU's. Using pk_{DA}, any SU is able to verify any received c*ert_{HN}* and extract the valid *pk_{HN}*.

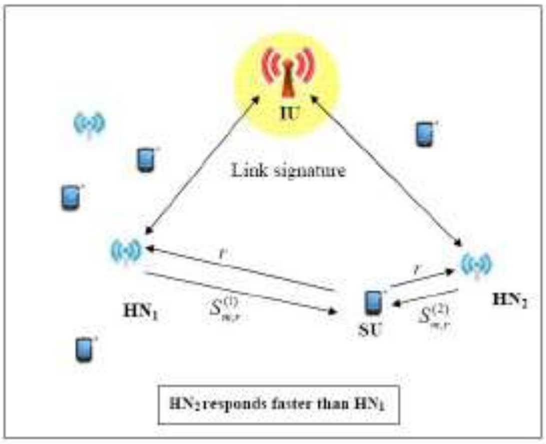

Figure 3. IU-HN and HN-SU interaction

IU-HN authentication phase: As we mentioned earlier, the HN authenticates the IU's using link (RF) signatures since employing cryptographic tools is impossible for this phase. To authenticate IU signals, a location distinction mechanism using multipath-based link signatures is employed. Authors in [16, 17], showed that a time-variant mutlipath fading channel between two fixed locations provides sufficient uniqueness to serve as a fingerprint of the fixed relationship between these locations. We briefly describe the link signature mechanism proposed in [13], in the context of IU-HN authentication. When IU_i transmits a signal *$s_i(t)$*, *HN_j* receives signal $r_j(t) = h_{ij}(t) * s_i(t) = \sum_{p=1}^{N} \alpha_p e^{j\phi_p} s_i(t - \tau_p)$ where $h_{ij}(t)$ denotes the impulse response of the unique channel between i and j, '*' denotes the convolution operation, and *N* denotes the number of distinct multipath components of *$s_i(t)$* at the receiver, each one delayed by α_p and phase-shifted by ϕ_p .

To obtain the desired impulse response, operations in the frequency domain yield $H_{ij}(f) = R_j(f)/S_i(f)$. To construct a link signature represented by *$H_{ij}(f)$* or *$h_{ij}(t)$*, the *$s_i(t)$* must be known at the helper. For this purpose, link signatures can be constructed using known sequences employed by the IU for control and channel estimators (usually implemented by adaptive filters and LMS estimators) on the helper's side. For more details the reader may refer to [13, 16, 17]. We assume that through RF signatures, the HN is able to accurately identify the IU's and is able to sense the spectrum white spaces (fallow bands). Thanks to the link signature protocols and their inventors.

HN-SU authentication phase: This phase operates in two rounds of communication as follows:

- SU: Prepares a fresh random nonce *r*.

- SU → HN: Broadcasts r to the nearest nodes. This random nonce is the challenge to the neighbors HN's to withstand the replay and wormhole attacks without the need to synchronize the network.
- HN: Given the tuple THN that was installed on the HN during deployment phase and the received r, HN runs the signing algorithm and prepares a signature, *sign(sk*$_{HN}$*, (m||r))* on *m* and *r*, where "||" denotes concatenation and m is the last spectrum information update of the IU's.

- HN → SU: Broadcasts the tuple,

$$S_{m,\ r} = \{(m||r), sign(sk_{HN}, (m||r)), cert_{HN}\ )\}$$

- SU: Receives tuples $S_{m,r}$ from the neighbor HN's. It then operates as follows:
  - Arranges them in ascending order of reception time, the first received is the fastest and is in the top of the list.
  - Starting from the top of the list, it uses pk_{DA} to extract the valid pk_{HN} from $cert_{HN}$ and uses pk_{HN} to verify the signature *sign(sk*$_{HN}$*, (m||r))*. If the signature is invalid, it ignores this tuple $S_m,_r$ and continue signature verification for the next in the list. Else; if the signature is valid, it accepts m as a valid message (but not necessarily a valid spectrum information).
  - Runs any trust evaluation algorithm to filter out false (authentic-but-malicious) reports.

6 SECURITY ANALYSIS

IU emulation attack. Given that the area (several meters) surrounding the IU is physically secured, the link signature between the IU and the HN prevents this type of attack.

Helper's message fabrication attack. In case the adversary is emulating a HN, elimination of this attack is straight forward assuming that the employed digital signature scheme is unforgeable. The weighted trust evaluation algorithm run by a secondary user as the last protocol phase filters out the reports received from a corrupted HN.

Helper's message replay attack. The adversary may record previously broadcasted spectrum information tuples S_{m^*,r^*} in an intention to replay them later to disrupt correctness. However, since the SU polls the HN's with a fresh random nonce $r^* \neq r$, which must be included in the signature on m, older tuples containing inconsistent nonce will fail the verification test and are immediately rejected by the SU.

Wormhole attack. As the reader may have noticed, from the description of the protocol, the main purpose of the random nonce *r* (by which SU challenges HN) is not to eliminate the wormhole activity. The adversary still able to relay messages from an SU in one area (say area A) to an HN in another area (say area B) and vice versa as shown in Figure 4.

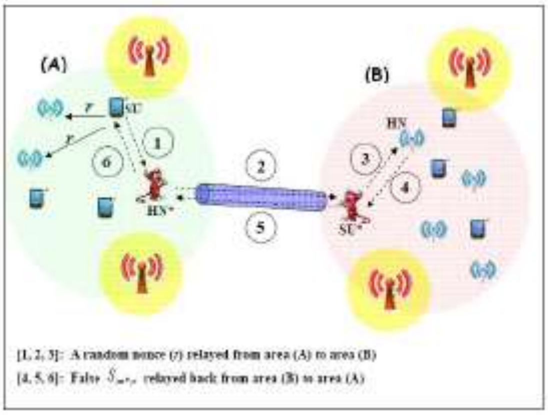
(A)
(B)
SU
r
r
1
6
HN*
2
5
3
4
HN
SU*
[1, 2, 3]: A random nonce (r) relayed from area (A) to area (B)
[4, 5, 6]: False $S_{m^*,r}$ relayed back from area (B) to area (A)

Figure 4. Execution of a single instant wormhole attack in our protocol showing its limited impact

However, with the random nonce provided by SU, the wormhole attack is now forced to delay. I.e. the adversary is not able to forge the signature *sign(sk*$_{HN}$ *, (m||r))* on r where r is generated by a particular SU since she actually does not know the secret key *sk*$_{HN}$. A simple replay of old messages from different areas will not work since the SU is expecting this particular fresh random nonce in the signature it receives from the HN.

Now, the adversary knows that SU will not simply accept a false spectrum information m_ unless he receives a valid signature on it concatenated with the currently broadcasted r with a valid secret key *sk*$_{HN}$ of any HN. The adversary's only way out is to run a wormhole attack in two-way of communication: From area A, she relays *r* (as if she is an SU) to any HN in area B. HN in area B will respond by a valid tuple $S_{m^*,r}$ where *sign(sk*$_{HN}$*, (m***||r))* is a valid signature on m^* and *r* using its secret key *sk*$_{HN}$.

An important remark we make, by comparing Figure 2 to Figure 4 is that: Figure 2 represents the execution of a wormhole attack in previous protocols that rely on periodic synchronous broadcast of spectrum information, notice that a single successful execution of wormhole has an impact on all secondary users in area B. On the other hand, a single execution of wormhole in our protocol shown in Figure 4, limits the attack impact to only one secondary user in area A which is inefficient for the adversary. Notice that not all SU's poll for spectrum information at the same time.

Moreover, the polling with random nonce forced the adversary to run the attack in two-way of communication on a link much larger in distance than a link between an SU and an honest neighbor HN. Notice that, the wormhole link between the two areas must be large enough for the attack to be effective or else, if they are too close, spectrum information relayed through the wormhole will not be different from honest ones and consequently ineffective. Hence, the responses from the honest HN neighbors will be much faster and occupy the top of the list of received signatures at the SU. From another point of view, allowing the SU to poll the neighboring HN's with the random nonce gives the SU a monitoring capability and control on the elapsed time until signed spectrum information is received (as if the SU has started a race) putting the HN's in a racing condition toward delivering the signature tuple $S_{m,r}$ as fast as possible to be at the top of the list. And since the wormhole attack in this case takes two rounds of communication, while an honest reply takes only one round and is much shorter in distance, a reply from an emulated HN is not likely to be at the top of the list.

Denial of service attacks (DoS). The strategy we proposed in this paper (polling helper nodes with random nonce in an asynchronous fashion) may give the adversary a chance to flood the helper nodes with fast random nonce's with high transmission rate, hopping to deny the service, reduce the performance of the helper nodes or at least delay their responses to give a chance for the wormhole attack to succeed. However, such attack may be tackled by allowing the helper nodes - when detecting a high traffic density or nonce's arriving with high rate (exceeding a certain threshold) - to run a location resolution algorithm to determine the location of the source(s) of the nonce. If the location of the source is the same for the received nonce's or very close in distance, the helper node simply ignores them and respond to the first nonce coming from a differently located source. We emphasize that the location resolution algorithm is run by a helper node and not a secondary user as mentioned in the contributions section of this paper.

Contacting single helper node. Another warning we have to bring attention to, is when the SU is in the situation where it is only able to communicate with one helper node and completely isolated from other nodes, although this is a very seldom situation. Of course, if this node is the emulation adversary, it definitely wins the race. In this case, the SU may either take the risk of accepting the spectrum information or takes a defensive action and ignore a single reception. We have to mention that there are plenty of nodes in the surrounding of the SU, hence, contacting one node and being completely isolated from other nodes is unlikely to occur.

7 SIMULATION AND RESULTS

In our proposed protocol, the packet delay of ad-hoc network plays a big role in distinguishing the honest helper nodes' replies from the wormhole attacker's reply. The packet delay on a specific one-hop link, denoted by D, can be divided into:

- Queueing delay: The interval between the time the packet enters in the queue of the link's emitter and the time that the packet becomes the head of line packet in this node's queue.

- Channel access delay: Time required to gain access to the channel. A popular mode of channel access is the CSMA/CA mode set by the IEEE 802:11 standard. Even though optimized channel access can be designed for specific ad-hoc networks, it is believed that the more common 802:11 access mechanism will be used to create a plug and play environment. As a result each node in the network uses a CSMA/CA protocol to access the channel.

- Mean transmission delay: Time between when the first and last bits are transmitted. This delay is proportional to the data rate and the packet's length in bits, It is given by the following formula:
DT = N / R

- Propagation delay: The time lag between the departure of a signal from the source and the arrival of the signal at the destination.

- Service time: The average time to service each Request. In our proposed protocol, it is the time required for signing the message ($m||r$).

In our proposed protocol, the transmitter has a timer but this timer will start to count after sensing and accessing the channel by sending the last bit of its request.
However, as the switchover to digital television frees up large areas between about 50 MHz and 700 MHz, we can allocate a channel for each adjacent helper node to speed up the response for multiple requests from SUs in their coverage area. Additionally, we assume that helper nodes don't have queues.

According to the first cognitive radio networking standard for personal/portable devices in TV white spaces, the lowest and the highest data rate modes are 4.75 Mbps and 23.74 Mbps, respectively. As the reply of helper nodes consisted of 3008 binary bits, (i.e. 376 bytes), assuming that the rate of transmission is 23.74 Mbps, hence the total transmission delay will take $1.26712*10^{-4}$ *sec* (126.7060 *μsec*).

Now, we need to compute the service time. However, as the cryptographic algorithms takes a significant amount of time if the algorithms are implemented in software, current advancements in technologies provide hardware cryptographic coprocessors for use in securing financial applications, e-commerce and SSL (Secure Socket Layer) transactions. These cryptographic coprocessors can perform 1250 DSA signatures per second and 620 DSA signature verifications per second [33]. Hence the service time will take $8 * 10^{-4}$ *sec*. To compute the propagation delay, it is required to determine the maximum distance between HNs and SUs. To guarantee a good quality the power of SUs should be in balance at the edge of the cell. The main idea behind the power budget calculations is to receive the output power level of SUs transmitter as a function of HNs sensitivity levels.

To determine the maximum distance between HN and SUs, we consider a ground reflection (two-ray) model for calculating the power level of a received signal over a distance, *d*. The received power level is given by [34]:

$$P_r(d) = P_t G_t G_r \frac{h_t^2 h_r^2}{d^4 L}$$

Where P_t is the transmitted power, $P_r(d)$ is the received power which is a function of the T-R separation, G_t, h_t are the transmitter gain and height, respectively, G_r, h_r are the receiver antenna gain and height, respectively, *d* is the T-R separation distance in meters, and *L* is the system loss factor not related to propagation ($L \geq 1$).

According to the FCC specifications, Personal/portable devices operations will be permitted at up to 100 mW EIRP, with no antenna gain. Figure 5 plots the received power at helper nodes versus the distance between SU and HN at 100mw EIRP at SU.

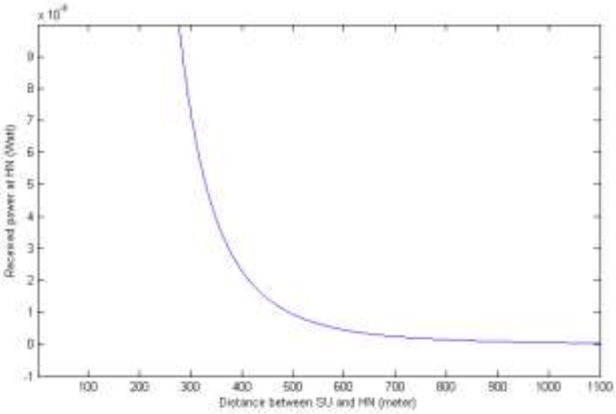

Figure 5. The received power at HN versus the distance between SU and HN

It is obvious that, at 1000 meter separation, the received power at HN will be $5.906*10^{-10}$ watt, which means that the HN sensitivity will be from -65 to - 70 dBm, which is a practical level. In the laboratory tests of TV signals, the Phase II prototype devices were able to detect a ‘‘clean,’’ i.e., unfaded, DTV signal on a single channel at levels in the range of -116 dBm to -126 dBm. The detection threshold sensitivity of the devices varied from -106 dBm to -128 dBm when recorded off-air DTV signals, which included multi-path fading and other ‘‘real-world’’ distortion, were used.

Finally, for a 1000m SU-HN maximum separation, we need to compute the propagation delay corresponding to the maximum distance which may separate between helper nodes and SUs (i.e. 1000 m)at the speed of light (c = *3* 10⁸ m/sec*). Figure 6 shows the relation between the distance and the propagation delay. It is obvious that the maximum delay at *1000m* away from HN is $6.667*10^{-6}$ sec.

The maximum acceptable delay for receiving the reply from honest helper nodes will be the sum of all previous delays. Hence:

D= $1.26712*10^{-4} + 8*10^{-4} + 6.667*10^{-6}$ *sec*

D= $9.3338*10^{-4}$ *sec*

As a result, SUs accept all replies of their latest request (with latest random nonce) within $9.3338*10^{-4}$ *sec* to be verified cryptographically and reject other delayed replies after this threshold.

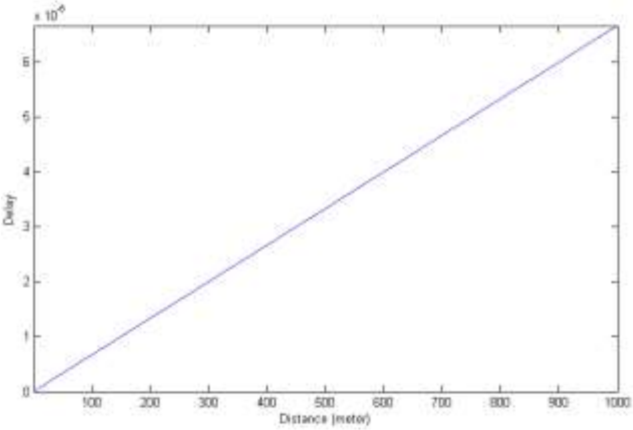

Figure 6. The distance in meter versus Delay

8 CONCLUSIONS

We proposed a protocol to allow a secondary user to efficiently authenticate helper nodes and consequently authenticate the spectrum information of the incumbent users. Our protocol does not require the helper nodes to operate in a synchronized fashion, each helper node operates completely in an asynchronous fashion independent of any other node in the network. The protocol withstands replay and wormhole attacks efficiently in the absence of helper nodes synchronization. We have completely analyzed the security and efficiency of our protocol. The protocol does not require the secondary user to store any secret information and uses a little memory to store a public key and several digital signatures. The computation complexity of the secondary user is limited to verifying several digital signatures and run a simple weighted trust evaluation algorithm. The helper nodes are located in close proximity to secondary users and hence, secondary users communicate with minimum power.

REFERENCES

[1] I. Akyildiz, W. Lee, M. Vuran, and S. Mohanty. NeXt generation/ dynamic spectrum access/cognitive radio wireless networks: a survey. Computer Networks, 50(13):2127-2159, 2006.

[2] H. Kim and K. Shin. In-band spectrum sensing in cognitive radio networks: energy detection or feature detection? In Proceedings of MOBICOM, pages 14-25, 2008.

[3] B. Wild and K. Ramchandran. Detecting primary receivers for cognitive radio applications. In Proceedings of IEEE DySPAN, pages 124 -130, 2005.

[4] BQ. Yuan, P. Tao, W. Wenbo, and Q. Rongrong. Cyclostationarity based spectrum sensing for wideband cognitive radio. In Proceedings of the WRI International Conference on Communications and Mobile Computing, volume 1, 2009.

[5] A. M. Wyglinski, M. Nekovee, T. Hou, Cognitive Radio Communications and Networks, ElSevier Inc., 2010.

[6] FCC. Second report and order and memorandum opinion and order, FCC-08-260, 2008.

[7] F. M. Salem, M. H. Ibrahim, I. I. Ibrahim, A Primary User Authentication Scheme for Secure Cognitive TV Spectrum Sharing. International Journal of Computer Science Issue, 9(4),pages 157-166, July 2012.

[8] F. M. Salem, M. H. Ibrahim,, I. I. Ibrahim, Energy Detection Based Sensing for Secure Cognitive Spectrum Sharing in the Presence of Primary User Emulation Attack. IEEK Transactions on Smart Processing and Computing, 2(6), pages 357-366, 2013.

[9] F. M. Salem, M. H. Ibrahim, I. A. Ali, I. I. Ibrahim, Matched-Filter-based Spectrum Sensing for Secure Cognitive Radio Network Communications. International Journal of Computer Applications, 87(18), pages 41-46, February 2014.

[10] S. Mishra, A. Sahai, and R. Brodersen, Cooperative sensing among cognitive radios, in Proceedings of the IEEE International Conference on Communications, June 2006.

[11] R. Chen, J. Park, and J. Reed. Defense against primary user emulation attacks in cognitive radio networks. IEEE Journal on Selected Areas in Communications, 26(1):2537, 2008.

[12] Y. Liu, P. Ning, and H. Dai. Authenticating primary users signals in cognitive radio networks via integrated cryptographic and wireless link signatures. In Proceedings of the 2010 IEEE Symposium on Security and Privacy, pages 286-301, 2010.

[13] S. Chandrashekar and L. Lazos. A Primary User Authentication System for Mobile Cognitive Radio Networks. In Applied Sciences in Biomedical and Communication Technologies (ISABEL), 2010 3rd International Symposium on (November 2010), pp. 1-5.

[14] S. Anand, Z. Jin, and K. Subbalakshmi. An analytical model for primary user emulation attacks in cognitive radio networks. In Proceedings of IEEE DySPAN, pages 1-6, 2008.

[15] Z. Chen, T. Cooklev, C. Chen, and C. Pomalaza-Raez. Modeling primary user emulation attacks and defenses in cognitive radio networks. In Proceedings of the 28th IEEE International Performance Computing and Communications Conference (IPCCC), pp. 208-215, 2009.

[16] N. Patwari and S. Kasera. Robust location distinction using temporal link signatures. In Proceedings of MOBICOM, page 122, 2007.

[17] L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe. Fingerprints in the ether: Using the physical layer for wireless authentication. In Proceedings of IEEE ICC, pages 4646-4651, 2007.

[18] M. Ben-Or, S. Goldwasser, and A. Wigderson, Completeness theorems for non-cryptographic fault tolerant distributed computation, ACM STOC 1988.

[19] R. Gennaro, S. Jarecki, H. Krawczyk, T. Rabin, Secure Distributed Key Generation for Discrete-Log Based Cryptosystems, Journal of Cryptology 20(1), pp. 51- 83, 2007.

[20] M. H. Ibrahim, Efficient Dealer-Less Threshold Sharing of Standard RSA, International Journal of Network Security,. Vol. 8, No. 2, pp. 139-150, 2009.

[21] M. H. Ibrahim, Resisting Traitors in Linkable Democratic Group. Signatures, International Journal of Network Security (IJNS), Vol. 9,. No. 1, 2009, pp. 51-60.

[22] Oded Goldreich, Secure multi-party computation, 4th ACM Conf. on Comp. and Comm. Security, 1998.

[23] R. Cramer , I. Damgard , S. Dziembowski , M. Hirt , T. Rabin, Efficient Multiparty Computations Secure Against an Adaptive Adversary, EUROCRYPT 1999, pp. 311-326.

[24] M. H. Ibrahim, A Novel Approach to Adaptively Secure Message Transmission in The Non-Erasure Model, International Journal of Advanced Research in Computer Science (IJARCS), Vol. 2, No. 6, Nov-Dec 2011.

[25] Wenkai Wang, Husheng Li, Yan Sun, Zhu Han, Attack-proof collaborative spectrum sensing in cognitive radio networks. In Proceedings of CISS'2009, pp. 130-134.

[26] Idris M. Atakli, Hongbing Hu, Yu Chen, Wei-Shinn Ku, Zhou Su, Malicious node detection in wireless sensor networks using weighted trust evaluation, SpringSim 2008, pp. 836-843.

[27] M. Mishra, A. Sahai, B. Brodersen, Cooperative Sensing among Cognitive Radios, ICC, June, 2006.

[28] Y. Hu, A. Perrig, and D. Johnson. Packet leashes: a defense against wormhole attacks in wireless networks. In Proceedings of INFOCOM, pages 1976-986, 2003.

[29] M. H. Ibrahim, Efficient Dealer-Less Threshold Sharing of Standard RSA. I. J. Network Security 8(2): 139-150, 2009.

[30] Saikat Basu, A New Parallel Window-Based Implementation of the Elliptic Curve Point Multiplication in Multi-Core Architectures, International Journal of Network Security, Vol. 13, No. 3, 2011, Page(s):234-241.

[31] R. L. Rivest, A. Shamir and L. Adleman, A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Communications of the ACM, 1978, pp. 120-126.

[32] Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, Tal Rabin: Robust Threshold DSS Signatures. EURO-CRYPT 1996: 354-371.

[33] SafeXcel-1841 Product Brief, SafeNet Inc., Belcamp, MD, 2005. Available:

[34] Theodore S. Rappaport. (2002). Wireless Communications principles and practice. Singapore: Pearson Education, Inc.

AUTHOR PROFILES

Maged Hamada Ibrahim received his BSc in communications and computers engineering from Helwan University, Cairo; Egypt, in 1995. He received his MSc and PhD in engineering cryptography and network security systems from Helwan University in 2001 and 2005 respectively. Currently, he is an associate professor since 2013 at Helwan University and also joining several network security projects in Egypt. His main interest is engineering cryptography and communications security. More specifically, working on the design of efficient and secure cryptographic algorithms and protocols, in particular, secure distributed multiparty computations, public key infrastructures, digital signatures, digital rights management protocols and non-cryptographic solutions to communication security problems. Other things that interest him are number theory and the inspection of mathematics for designing secure and efficient cryptographic schemes.

Fatty M. Salem Received her BSc degree in electronics, communications and computers engineering from Helwan University, Cairo, Egypt, in 2007. She received her MSc and PhD degree in network security from Helwan University, in 2010 and 2014 respectively. Her main interests include systems, cryptography, and network security. Currently, she is a lecturer since 2014 at Helwan University.