Authenticating Hard Decision Sensing Reports in Cognitive Radio Networks*

Helena Rifà-Pous[a,*], Carles Garrigues[a]

[a]*Internet Interdisciplinary Institute, Universitat Oberta de Catalunya, Rb. del Poblenou 156, 08018 Barcelona, Spain*

Abstract

Cognitive radio networks sense spectrum occupancy and manage themselves to operate in unused bands without disturbing licensed users. Spectrum sensing can be more accurate if jointly performed by several nodes. In order to get a successful result, avoiding fake nodes' inputs is required and so, it is necessary to authenticate their local sensing reports. A few authentication algorithms have been proposed up to now. However, they introduce a notable overhead in lightweight hard decision systems. In this paper we present an efficient protocol based on symmetric cryptography and one-way functions, and an analysis of its security features. The system allows determining a final sensing decision from multiple sources in a quick and secure way.

Keywords: authentication, cognitive radio, cooperative sensing, hard fusion, security

1. Introduction

Spectrum is an essential resource for the provision of mobile services. In order to control and delimit its use, governmental agencies set up regulatory policies. Unfortunately, such policies have led to a deficiency of spectrum as only few frequency bands are left unlicensed, and these are used for the majority of new emerging wireless applications. Besides, studies conducted by the Spectrum Policy Task Force show that most of the licensed spectrum is largely under-utilized [1].

One promising way to alleviate the spectrum shortage problem is adopting a spectrum sharing paradigm in which frequency bands are used opportunistically. In this scheme, those who own the license to use the spectrum are referred to as primary users, and those who access the spectrum opportunistically are referred to as secondary users. Secondary users must not interfere with primary ones, who always have usage priority.

The enabling technology for opportunistic sharing is cognitive radio (CR) [2]. A CR is a system that senses its electromagnetic environment and can dynamically and autonomously adjust its operating parameters to access the spectrum. CR terminals form self-organizing networks capable to detect vacant spectrum bands that can be used without harmful interference with primary users. Once a vacant band is found, secondary users coordinate themselves in order to share the available spectrum.

Performing reliable spectrum sensing is a difficult task. Wireless channels can suffer fading, thus provoking the hidden node problem in which a secondary user fails to detect a primary transmitter. The most important challenge for a CR is to identify the presence of primary users, and, for this reason, secondary users must be significantly more sensitive in detecting primary transmissions than primary receivers.

In order to reduce the sensitivity requirements of individual CRs, recent studies propose performing distributed spectrum sensing (DSS)[3]. In DSS, multiple secondary users conduct local spectrum sensing. Then, their results are merged to reach a final decision. Several data fusion schemes have been proposed to merge the sensing data observed by each secondary user. These schemes are based on exchanging of more or less information depending on whether devices perform hard or soft cooperation. When hard cooperation is employed, radios only exchange their final decision: primary user detected or not detected. On the other hand, soft decision means that radios exchange their local measures and/or test statistics with each other. Among the proposed methods, the most typical one is based on applying the "k out of N" rule. This rule determines that the channel is occupied if at least k of the N secondary users have detected the primary signal. As avoiding interference with primary users is a top priority, the most common value of k is 1.

In order to correctly balance the contributions of the users and ensure a reliable result, data fusion algorithms try to characterize the users, learn how they behave and to what extend they shall be trusted, using either probabilistic [4, 5, 6] or reputation models [7, 8]. However, in order to effectively track users, the sensing contributions that they make must be authenticated. Some proposals have been presented to authenticate users' spectrum decisions using message authentication codes (MAC) [9, 10]. Even though the protocols are provably secure, they are not fitted for hard decision techniques due to the huge data overhead introduced in the reports. If the number of nodes participating in the sensing process is high, the amount of information needed to transmit the sensing information perceived by every node

*This work is partially supported by the Spanish Ministry of Science and Innovation and the FEDER funds under the grants TSI-020100-2009-374 SAT2, TSI2007-65406-C03-03 E-AEGIS and CONSOLIDER CSD2007-00004 ARES

*Corresponding Author

Email addresses: hrifa@uoc.edu (Helena Rifà-Pous), cgarrigueso@uoc.edu (Carles Garrigues)

along with its MAC signature is very large. Therefore, a more lightweight protocol is needed that allows for the large number of network nodes that can form a CR network.

This paper presents a protocol that enables the secure authentication of hard decision sensing reports in an efficient way. Reports can be authenticated as soon as they reach the fusion centre, and the process is simple and fast since is basically based on hash functions.

2. Protocol

This section presents our protocol for the secure authentication of users' sensing reports. The protocol prevents users from illegitimately claiming false identities and from injecting fake sensing data. Thus, the protocol aims at withstanding the following attacks:

- Altering the final sensing decision. A user could increment her weight in the data fusion process by forging several identities and making a contribution for each of them. With enough forged identities, a user might be able to completely alter the aggregate reading.
- Deceiving the reputation system. By using a different identity each time, a user might report false sensing data repeatedly and avoid earning a bad reputation.
- Obtaining resources unfairly. A user could use many identities to obtain more than her fair share of resources (e.g. bandwidth).

The proposed protocol assumes that the cooperation among CR's is implemented in a centralized manner, which is the most frequently used configuration in the spectrum sensing protocols presented to date. The protocol is designed for hard cooperation schemes, as they ensure that the amount of information sent through the network is minimal. We also assume that the exchange of messages between the secondary users and the fusion centre is carried out through a common control channel. The mechanism used to implement this control channel is out of the scope of this paper. Nodes are all loosely synchronized with the fusion centre up to some time synchronization error Δ, where Δ is the maximum network propagation delay plus the maximum time synchronization error.

To perform distributed sensing securely, the cooperative system should identify the users that participate in the sensing process, authenticate their claims, and weigh up their contribution to the final decision based on their reputation or probability of successful detection. Our protocol focuses on the mechanisms required to identify the users and authenticate their sensing results. The final part of the distributed sensing process (i.e. weighing up and merging the contributions) can be implemented using any of the mechanisms that we have mentioned in the previous section. The selection of which data fusion technique to use is out of the scope of this paper.

The protocol is divided in three phases. The first phase is the registry of users; the second one the sensing assignment; and finally the third phase is the collection of sensing results. In the following sections, we will start describing the security framework we use to provide security services and then we will explain each of the protocol phases in detail.

2.1. Security Framework

One of the key goals of the protocol design is to develop an efficient solution suitable for constrained devices. Therefore, the cryptography involved in our proposal is based on simple hash functions and symmetric keys.

The use of symmetric keys is essential to implement the authentication of the sensing data provided by the secondary users. However, the main challenge of symmetric key systems is how to distribute and manage the keys among the authorized nodes. Different lightweight processing solutions have been proposed in the scope of sensor networks that pre-distribute or dynamically generate the secret keys using probabilistic approaches (see a review in [11]). However, such schemes are impractical for CRNs due to the particular features that differentiate a CRN from a traditional sensor network, namely:

1. The topology of CRNs is continuously changing. Some sensor networks are dynamic in the sense that they allow addition and deletion of sensor nodes after deployment to extend the network or replace failing and unreliable nodes without physical contact; however, the dynamism of CRN goes further: nodes are mobile and join and leave a particular community in short periods of time.
2. The number of network members is several orders of magnitude larger than that of sensor networks. The number of connected members in a particular moment is similar to a sensor network, but in an open CRN network, the number of potential users is unlimited and so, key management must be highly scalable. Moreover, it must allow the addition of new users in the system in the course of time as opposed to admitting them all at once at system start-up.
3. In a CRN the channels can only be used for limited periods of time (while primary users are not active). So, the time available for data transmission must be maximized and the security protocols must be designed in such a way that they introduce the minimum possible overhead.

With these challenging operational requirements, the use of sensor network designed schemes for key distribution becomes too complex in CRNs.

Taking advantage of the fact that CRNs are suitable for more powerful devices than sensor networks are, we take a public key infrastructure (PKI) approach to initialize the network. We conceive the fusion centre as a well-known and static entity in the network that manages the spectrum of a certain area and connects to the Internet and to any Validation Authority (VA), if required. The role of the fusion centre can be assumed by a secondary base-station or a spectrum broker, which are entities found in most well-known CR network architectures, such as IEEE 802.22 [12] or DIMSUMNet [13].

The fusion centre must hold a certificate that is made available to the users from different means (web, public directory, etc.) and the users must know and be able to validate it. Likewise,

users must hold a valid certificate from a recognized Internet Certification Authority (CA).

Peers are first authenticated through digital signatures and, at the same time, they commit to an identification key hash chain. Even though the operation of generating a digital signature is not light, users only have to do it once, in the setup phase, and so it is totally assumable [14]. Then, we make use of identification keys and efficient one-way functions to protect users' sensing reports from forgery and manipulation. Identification keys are taken from a two-dimensional key chain consisting of a high-level (primary) chain and multiple low-level (secondary) chains. Low-level chains provide evidence on the nodes' sensing reports, while the high-level chain is employed to generate one-time HMAC signatures that endorse users' commitments and to broaden the authenticity of users' claims to the low-level chain.

Different one-way functions may be used for high-level and low-level chains with the aim of better dealing with the trade-off between security and efficiency. In the proposed protocol, both high-level and low-level chains are based on cryptographic hash functions, such as SHA-1.

Hash Chains, first proposed by Lamport [15], are versatile low-cost constructions that are used extensively in various cryptographic systems. A hash chain of length N is constructed by applying a one-way hash function H(.) recursively to an initial seed value v_N: $v_{N-1} = H(v_N)$, $v_{N-2} = H(v_{N-1})$, $\cdots$, $v_0 = H^N(v_N)$. In general, $v_i = H(v_{i+1}) = H^{N-i}(v_N)$. The last element of the chain v_0 is called the top value.

The high-level chain used in the protocol is a generic hash chain with N elements, from $\{V_0 \cdots V_{N-1}\}$. Its elements will be used at irregular times during a large time frame. On the contrary, low-level chains are shorter (they have m elements, with $m < N$) and their elements will be employed periodically, but only for short periods of time. Since low-level chains are short lived, they are less demanding regarding security requirements than high-level chains. So, we construct them with truncated short hash values in order to reduce the bandwidth overhead of their transmission.

Low-level chains are introduced in the system through high-level chains. Each element of a high-level chain in used to commit, employing an HMAC, to the top element of the low-level chains. The index of the high-level chain element used to sign the low-level chain, is the first number of the tuple that identifies the low-level elements (as we will see next).

Besides, to strengthen the relation between high-level and low-level chains even more and avoid that randomized chains from intruders can be accepted as authentic, we use an element of the high-level chain as a seed to create low-level chains. Low-level chains that are introduced in the system with a signature that uses the high-level element V_i, are created from the seed provided by the V_{i+1} element, and have the structure:

$$\{V_{i,0,0} \cdots V_{i,0,m}\}, \{V_{i,1,0} \cdots V_{i,1,m}\}, \cdots, \{V_{i,c-1,0} \cdots V_{i,c-1,m}\}$$

where c is the number of generated low-level chains.

The use of short valued chains may introduce pre-computation attacks. To avoid them, low-level chains are computed appending a salt s to the input value of the hash function in each step of the algorithm. This salt is a network public element that is revealed just before the use of low-level chains. In particular, the salt is the element of the high-level chain that is used to commit to the low-level chain. So, for creating a generic chain $V_{i,c,m}$, the seed V_i is employed.

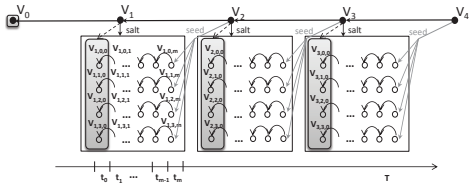


Figura 1: Two-dimensional key chain. The top values of the chain are highlighted in grey, and are the ones that are first published. The values of the chain are disclosed in the reverse order of their generation.

Figure 1 shows the architecture of the two-dimensional key chain. The arrows indicate the generation order of the elements and the dependencies between them. The slashed line signals that the values of the high-level chain are used to commit to the top values of the low-level chains. It is worth noting that this set of hash chains is not generated in just one go. As will be seen, the low-level chains associated with each root element are only generated when they are needed to sign more sensing results.

2.2. Protocol Phase 1: User Registry

In the first phase, the user contacts the fusion centre (which can be, for instance, the base station) and asks permission to join the cognitive radio network. Besides, she commits to a two-dimensional key chain by attaching the top value of the high-level chain in the request. This process requires mutual authentication using digital signatures. At this point, the fusion centre decides whether or not to accept the user into the network. The following are the detailed steps carried out during this phase.

1. User U chooses a random number V_N and prepares a high level chain of length N, where N is chosen by the user according to its memory resources.
2. U sends the top value of her chain (V_0) to the fusion centre FC in a digitally signed message. The signature is computed using U's private key pvk_U. She also includes information about her identity Id_U (i.e. the unique identifier of her public key certificate).

$$JoinReq = \{V_0, Id_U, Sign_{pvk_U}(V_0, Id_U)\}$$

3. FC verifies the signature received from U using U's public key pbk_U. If the signature is correct, FC decides whether or not to accept U into the network. This decision will be based, for example, on the reputation earned by U in previous processes. The implementation of these mechanisms is out of the scope of this paper.
   If user U is accepted in the network, FC stores her identity Id_U, her MAC-layer address (it will be used in the sensing phase to identify the node), and her top chain value V_0 in a table.

2.3. Protocol Phase 2: Sensing Assignment

In the second phase, the fusion centre requests each user to sense a certain set of frequency bands by the submission of a public key digital signed message ($SensReq$). If users accept to sense the requested bands, they respond with a message in which they bound to a set of low-level key chains, two for each channel they are alloted. The authenticity of user's messages is ensured by a symmetric digital signature. Symmetric digital signatures are generated using a Hash Message Authentication Code (HMAC) function. HMACs provide message authenticity and integrity by calculating a hash of two inputs: the target message and a secret key. In our protocol, secret keys are taken from the pre-computed high-level hash chain. Here is a sketch of the Sensing Assignment Phase approach:

1. At time t_0, FC splits up the time into equal length intervals t_s and broadcasts a signed message with information about the schedule they will use in the sensing process, and a task list ($TaskList$) that contains the list of channels each user has to sense.

$$SensReq_{t_0} = \{TaskList_{t_0}, t_0, t_s, m, \\ Sign_{pvk_{FC}}(TaskList_t, t_0, t_s, m)\}$$

where

$$TaskList_{t_0} = [\ (Id_0, ChannelList_0, i_0) \cdots \\ (Id_S, ChannelList_S, i_S)\ ]$$

In the above expression, S is the total number of secondary users, i_j is the high-level chain index that points to the value the user j must use in the following step, and m is the length of the required low-level chains, as well as the number of times a channel must be sensed.

2. Each user U verifies the signature of the sensing request and, if correct, forms two one-way chains for each channel she is requested to sense: one chain is associated to an *empty* decision, the other is associated to a *occupied* decision.
These one-way chains are constructed from the high-level chain of each user as explained in section 2.1. In particular, the generation of low-level chains linked with the high-level ith position is as follows:

   - The generator computes a hash of the concatenation of three values: V_{i+1}, V_i, and x, for each x from 0 to $c-1$, where c is the number of chains that the user needs to create (that is, two chains for each requested channel). The result is truncated to 64 bits and assigned to the last value m of a secondary chain. Denoting $T(d,b)$ the truncation of some data d to the b leftmost bits, and $||$ the concatenation function, the operation can be summarized as:

$$V_{i,x,m} = T(Hash(V_{i+1}||V_i||x), 64)$$

   - The chain values, from $\mu = 0$ to $\mu = m-1$, are generated from the previous output of the function and the value V_i of the primary chain, which is used as a salt.

$$V_{i,x,\mu} = T(Hash(V_{i,x,\mu+1}||V_i), 64)$$

U assigns the generated keys sequentially to time intervals (one key per time interval). The one-way chain is used in the reverse order of generation, so any key employed in a particular time interval can be used to derive keys of previous time intervals, but gives no information about subsequent keys.
Figure 2 shows an example. U is requested to sense two channels and return the commitment signed with the high-level key V_1. She is asked to generate chains of length $m = 4$, which means that the channel will be sensed four times. She generates 4 chains with top values $\{V_{1,0,0}, V_{1,1,0}, V_{1,2,0},$ The subsequent values of the low-level hash chains $\{V_{1,0,\mu}, V_{1,1,\mu},$ are scheduled to be used in the time frame $t_\mu = t_0 + \mu \cdot t_s$, with $\mu = 1 \cdots 4$.

3. U publishes the top values of her light hash chains using an HMAC with her ith key of the high-level chain. U sends this message in the first time interval defined by the FC.

$$SensCommit_{t_0} = \{Id_U, ChCommit, \\ HMAC_{V_i}(Id_U, ChCommit)\}$$

where

$$ChCommit = [V_{i,0,0}, V_{i,1,0}, .., V_{i,c-1,0}]$$

4. FC receives $SensCommit_{t_0}$ and stores this information associated with user Id_U for a latter validation.

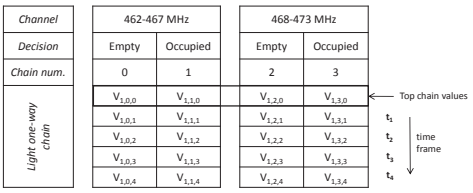


Figura 2: Example of light chains generated to report the sensing results of two channels. Chains are authenticated using key V_1 and are derived from key V_2

2.4. Protocol Phase 3: Collection of Local Sensing Results

In the third phase, users conduct spectrum sensing using a mechanism based on the energy perceived, cyclostationary statistics, or any other method, and take a decision whether a channel is occupied or not. These decisions designate which low-level chain has to be used to encode the sensing result of a spectrum band, from the two possible chains linked with each channel. Users publish their results using the elements of the selected low-level chains that are scheduled for the current interval. When the fusion centre receives the reports, it can verify they are authentic and integer since the legitimate user is the only one who has enough data to reveal the hash chain values associated with the present time frame. Users can only send one sensing report in each time interval.

Figure 3 shows an example of the elements of the two-dimension chain used and revealed during the Sensing Assignment phase and the Collection of Local Sensing Results phase. User U is requested to sense a channel and she commits to her two low-level chains computing an HMAC with the high-level

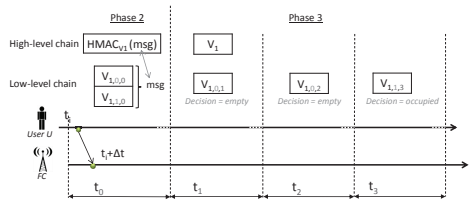


Figura 3: Chain information revealed by a user during the Sensing Assignment and Publication of Sensing Results. The user senses a single channel. The two low-level chains associated with this channel are authenticated using key V_1.

V_1 key. In the time frame t_1, U reveals V_1 so that the fusion centre can validate her commitment. U also sends the sensing decision for that period of time. In each interval, U reveals the value of one of her chains ($V_{1,0,\mu}$ or $V_{1,1,\mu}$) according to her decision in the present time frame t_μ. Users must try not to send reports in the last moments of a time frame to avoid that synchronization errors and transmission delays cause the reports to be dropped. In any case, the total synchronization error is much lower than the time of an interval ($\Delta \ll t_s$), so the problem is minimal.

The following are the detailed steps carried out in this phase.

1. In the time interval t_μ, each user U senses the channels listed in $ChannelList_U$ received in $SensReq$ message. After completing the sensing process, each user sends the results $SensRes$ to FC. These results are binary decisions since the protocol uses hard cooperation. To allow the authentication of the sensing results, these results are sent as follows:

$$SensRes_{t_\mu} = \{ChannelDes_1(\mu), \cdots, ChannelDes_{c/2}(\mu)\}$$

where

$$ChannelDes_x(\mu) = V_{i,x,\mu}$$

In the above expression, i is the index received from FC in the $SensReq_{t_0}$, c is the number of hash chains that the user has created (two for each channel requested to sense), and x is the low-level hash chain associated with the sensing decision of a particular channel. Note that the user do not send her Id_U in the message since she is identified using the MAC layer address.
Besides, in the time interval t_1, U also sends the ith element of her high-level chain (V_i) along with the report sensing message.
2. When FC receives V_i, it checks whether the key is correct and then checks the correctness of the buffered $SensCommit$ message, which was sent in the second phase of the protocol (in time frame t_0). Because FC knows the authentic key V_{i-1}, it can verify the authenticity of V_i by checking that $H(V_{i-1})$ equals V_i. FC knows the schedule for disclosing chain values and, since the clocks are synchronized, it can verify that the received value is still secret.
3. In each time interval t_μ, FC waits for the $SensRes_{t_\mu}$ of all secondary users. For each received message, it verifies the authenticity of the sensing response $V_{i,x,\mu}$ by checking that $T(Hash(V_{i,x,\mu}||V_i), 64)$ equals $V_{i,x,\mu-1}$. Note that FC may need to repeat this operation p times if the last known value of the hash chain is $V_{i,c,\mu-p}$. If the received data is correct, FC stores this data to optimize the validation of subsequent values of the chain, and with all data received, it starts the fusion of the sensing results to determine the joint decision.

The sensing process is repeated until the FC sends a message to stop it, or until exhausting the elements of the low-level hash chain. Then, the protocol returns to the second phase and FC generates a new sensing request. Figure 4 summarizes the messages exchanged between a user and the FC.

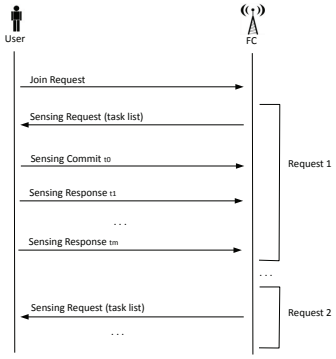


Figura 4: Diagram of the protocol messages

3. Discussion

In this section, we discuss the efficiency and security of the proposed protocol.

3.1. Temporal Overhead

The presented protocol provides a way to authenticate sensing reports with a minimum overhead. Each user has to generate a digital signature when she accesses the fusion centre for the first time. Afterwards, she only has to validate digital signatures and perform hash operations.

The presented protocol is designed for an open network in which cooperative sensing is performed with the users that are active and close to one another in a particular moment. As a result, the group of users collaborating in the sensing is very dynamic. The challenge of using hash chains to provide security in this scenario is that the participation of the users in the sensing tasks is very irregular. Thus, if we used a protocol where only a single chain was used, the overhead of storing and computing lots of elements of different chains (which might not be used afterwards) could become too high.

If a single key chain was used, the values of this chain would be used for the validation of every single sensing result. Therefore, this chain would quickly become very long, which is not a problem unless the users participate in the sensing process only sporadically. In this case, the computational cost of validating a chain element v_i from a distant preceding element v_j (with

$j << i$) would be very high. Although some mechanisms for fast hash chain traversal [16] and for economic setup and verification [17] have been proposed, the cost to access an element of a hash chain of length N is still log(N) in computations and log(N) storage. Considering a high number of users having an irregular presence in the network, this validation time would easily become unaffordable.

On the other hand, if a two-dimensional key chain is used, the validation of the sensing results is carried out from a low-level key chain, which has a very limited length because it is only used in a single sensing request. As a result, the validation times are small even if the CRN has a high number of irregular users.

In order to adapt to this dynamic context, we propose that users have a high-level chain whose elements are used on demand. The fusion centre has a counter for each user, and when this user wants to collaborate, she has to authenticate herself using a specific element of the chain. The authentication procedure is performed using an HMAC function, which is a two step authentication. Firstly, the user sends the HMAC signature, and afterwards she reveals the HMAC key so that the recipients can verify its authenticity. Although the protocol needs loosely time synchronization, no values of the high-level chain are wasted.

Unlike high-level chains, low-level chains are scheduled to be used in a fixed time frame. This is not a problem since users commit to low-level chains when they are active in the network and ready to collaborate. Moreover, these chains are short and they are only used for a particular sensing assignment.

Table 1 depicts the computational and transmission costs of phases 2 and 3 of the protocol in a network with 50 active users. We assume the sensing interval is $t_s = 2s$, and the length of low-level chains is $m = 500$. Thus, the maximum lifetime of phase 3 is 16,6 minutes. Then, low level chains have to be renewed.

The fusion centre triggers the sensing process by sending a $SensReq$ message. $SensReq$ contains a task list for each active member of the network. If each user is assigned 2 channels, the task list has a size of 4 bytes per user. Besides, if the contents of the message are signed using an 1024-RSA key, the approximate size of $SensReq$ is 350 bytes.

The packet transmission time T_{packet} over a control channel, which we assume to be a IEEE 802.11b network, is expressed as follows:

$$T_{packet} = T_{PhyHdr} + (M_{MacHdr} + M_{Payload})/11Mbps$$

where T_{PhyHdr} is the PLCP (Physical Layer Convergence Procedure) preamble and header. The physical control data is 24 bytes long and it is transmitted at 1Mbps, so $T_{PhyHdr} = 192\mu s$. M_{MacHdr} is the length of layer 2 headers, which for an ad-hoc connection is 24 bytes. $M_{Payload}$ is the protocol data length. Both M_{MacHdr} and $M_{Payload}$ are transmitted at 11Mbps. Then, transmitting a $SensReq$ message of 350 bytes over a 802.11b network takes nearly $0,5ms$.

Assuming FC is hosted on a AMD Opteron/2,2GHz processor running Linux, the signature generation time is $0,67ms$ [18].

Cost	Agent	Protocol	Operations	Time
Computation	FC	SensReq	PK Signature	$670\mu s$
		SensRes	2 hash·ch·user	$15\mu s$
	User	SensReq	PK Verification	$2,72ms$
		SensCom	4 hash chain + HMAC	$10ms$
Transmis	FC	SensReq	data + signature (350B)	$464\mu s$
	User	SensCom	data + HMAC (100B)	$283\mu s$
		SensRes	2 ch_elements (16B)	$221\mu s$

Cuadro 1: Analysis of time complexity of Sensing Assignment and Collection of Results

For a user that has an embedded ARM device at 624MHz, the costs of receiving and processing a sensing assignment of 2 channels are the following [19]:

- validating an RSA-1024 signature of the $SensReq$ costs $2,72ms$
- generating a $SensCommit$ comprises two steps: (1) building 4 hash chains based on SHA-1 and with a length of 500 elements takes $10ms$ (one hash operation is $5\mu s$), (2) generatating an HMAC to commit to the top values of the low-level chains, which involves two hash computations, takes $10\mu s$.

Besides, the $SensCommit$ message has a total length of 100 bytes and, as a result, sending it through a 802.11b network takes $283\mu s$.

A user can chose between storing the elements of the 4 hash chains she has computed, or recomputing the required elements every time she needs to send a sensing report. She will weigh the options considering her available resources. In this example, storing 4 hash chains represents 16KB of memory.

When the user sends a sensing report in a $SensRes$ message, she has to include one low-level chain element per sensed channel in the packet. The payload size of a user that senses 2 channels is 16 bytes, and takes $221\mu s$ to transmit it. In order to validate the sensing reports, the FC has to compute an average of 2 hashes per user channel. In a network with 50 active users, the FC should calculate 200 hashes in each time slot with a time cost of approximately $15\mu s$.

Taking into account the overall cost of phases 2 and 3 of the protocol, we conclude that during the duration of the sensing process (16,6 minutes) a node spends $12,72ms$ performing cryptographic computations (around $1,28{,}10^{-7}$ % of the time). The time required to send all sensing agreements and reports occupies the network for $5,56s$ (around $5,59{,}10^{-5}$ % of the time), which represents an overhead of $5,66{,}10^{-4}$ % compared to a plain sensing system without security.

In table 2 we compare the time complexity per node of our protocol with a plain scheme without security, with the secure schemes based on MACs proposed in [9] and [10], and with the straightforward solution of providing authenticity by digitally signing the sensing messages. The notations in the analysis are

defined as follows (the values in brackets are the default values -extracted from [19]- used to make the computations):

- T_H : Time to compute a hash function *(SHA-1:* $5,02\mu s$*)*
- T_S : Time to sign with an asymmetric key *(RSA-1024:* $24,05ms$*)*
- T_V : Time to verify with an asymmetric key *(RSA-1024:* $2,72ms$*)*
- T_E : Time to encrypt using a block cipher *(AES-128:* $80,73\mu s$*)*
- T_i : Time to transmit initialization data (e.g. sensing assignment phase)
- T_k : Time to transmit a key to open a commitment
- T_d : Time to transmit a sensing packet of length L_d
- L_d : Length of the data of a sensing packet (bytes)
- m : Number of sensings requested by the FC (length of hash-chains)
- n : Number of sensings actually carried out

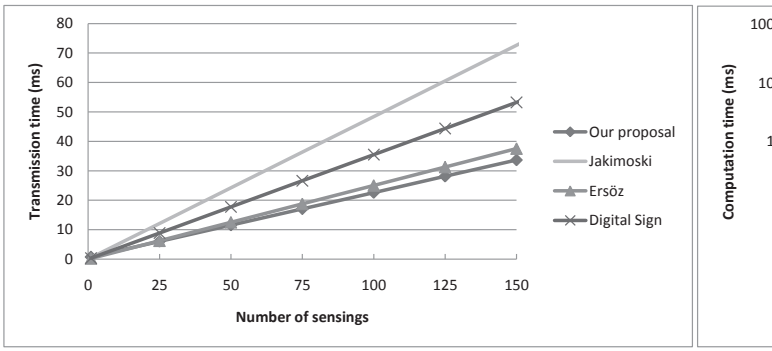
Schemes	Transmission		Computation
	L_d	Time (ms)	Time (ms)
No security	1	$n \cdot T_d$ = 105,09	-
Our proposal	16	$T_i + T_k + n \cdot T_d$ = 111,04	$T_V + 4 \cdot m \cdot T_H$ = 12,72
Jakimoski [9]	45	$n \cdot (T_k + T_d)$ = 242, 18	$T_V + m \cdot (T_E + T_H)$ = 45,59
Ersöz [10]	56	$n \cdot T_d$= 125, 09	$T_V + T_E + 2 \cdot n \cdot T_E$ = 83,53
Digital Sign	200	$n \cdot T_d$ = 177, 45	$T_V + n \cdot T_S$ = 12027,72

Cuadro 2: Comparison of time complexity for some schemes

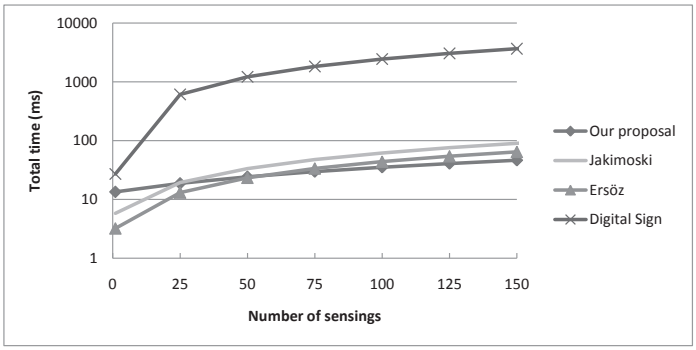
Table 2 shows the generic transmission and computational costs of the channel assignment and sensing phase of a CRN that assigns each user 2 channels to sense in a hard decision fashion for a set up m of 500 sensings. The table also depicts the particular costs of a network with the characteristics stated for computing table 1 and for a total period n of 500 sensings. Note that n can be less than m if users abandon the CRN or the FC requests secondary users to stop the sensing.

The results show that our proposal is the most efficient secure scheme in terms of both computation and transmission. The Jakimoski protocol presents transmission rates remarkably higher than the other protocols. This is because for each sensing report the user needs to send two packets to the fusion centre: (1) the sensing report and its MAC, (2) the key used to generate the MAC. This feature also introduce a notable delay in taking a final decision about the occupancy of a channel. In contrast, the sensing related costs of the Ersöz proposal are quite restricted, but it can present high management costs in quite dynamic networks since the keys are based on a Logical Key Hierarchy (LKH) architecture. In LKH users share a common key that has to be changed and re-distributed when the group of cognitive radio nodes is modified.

The costs of the proposed protocol are optimized when users stay in the network for periods of $n = k \cdot m$ sensings, with $k \in \mathbb{N}$, since the computational focus of the algorithm is at the beginning of each m period when low level chain elements are generated. In figure 5 we compare the costs of the analysed schemes for sensing periods $n < m$, with $m = 500$. Graphics show that for $n \leq 55$ the best algorithm is Ersöz, but when the number of sensings exceeds 55, our proposal is the most efficient one.



(a) transmission



(c) total *(log scale graph)*

Figura 5: Comparison of time complexity over the number of sensings n when $m = 500$

3.2. Security Analysis

From the security point of view, the proposed system is robust against Sybil attacks, in which a user illegitimately claims multiple identities, and the injection of false sensing reports. Sybil attacks are prevented using certificates generated by a trusted central authority. If a user does not own a valid certificate, she is not authorized in the CR network and cannot send sensing reports to the fusion centre. On the other hand, the injection of false sensing reports is avoided using verifiable reports, which are based on the use of one-way chains.

Cryptographic hash functions provide the required security properties to create a robust one-way chain. The main security threats for hash chains are their vulnerability against birthday attacks and hash chain attacks.

- The birthday attack is based on the birthday paradox, which states that the probability that two or more people in a group of 23 share the same birthday is greater than 50 %. This paradox can be mathematically applied to hash functions. If an attacker selects a random seed and computes a hash chain from this, the chances of a collision between any of the computed values and a value in an existing chain increase with the length of the chain. This can be avoided concatenating the index of the hash value in the chain when computing the hash. For a

one-way chain value of m bits, the expected cost for an attacker to find a pre-image or even a second pre-image is 2^{m-1} hash computations. Although not explicitly indicated, all the hash operations performed in the proposed protocol are computed using this index.

- The hash chain attack occurs when the hash chain contains a cycle, and thus, all the elements on the cycle stop having the one-way property. Furthermore, if the chain enters the cycle from an external value, it means there is a collision and this can result in a serious security attack. A recent study of hash chain vulnerabilities [20] indicates that the use of commonly used hash functions is pretty secure against hash chain attacks with probabilistic algorithms.

Reporting verifiable sensing results involves three steps. First the user sends the top elements of the low-level hash chains and the corresponding HMAC. Then, she reveals the HMAC key, which is an element of the high-level hash chain. With this information, the fusion centre verifies the integrity of the sensing commitment and the authenticity of the key. Finally, the user sends the sensing reports revealing some elements of the low-level chain she has just committed to.

The message that carries the sensing commitment is protected against modification attacks since it is signed with an HMAC. Keys used to compute the HMAC are taken from the high-level hash chain V of each user. If we use a chain based on SHA-1, then the security of HMAC operations is 112 bits, which meets the NIST security recommendations [21].

Replying attacks against sensing commitments are avoided because each HMAC key is used only once and because the $SensRes$ message carries the present time. In the sensing request, the fusion centre indicates which element i has to be used. Chain elements are requested in ascending order (V_1,V_2, $\cdots$ V_N), so knowing a user's previous key gives no information about the present one. Besides, the sensing request is signed so that an attacker cannot modify the requested hash index.

Additionally, as the sensing requests and commitment replies are synchronized by the index i, it is not effective to block the user's reports in order to steal her keys to later generate fake reports.

In the same way, sensing reports are also securely protected. They are transmitted as the elements of the low-level chains, and these low-level chains are bound to a high-level chain (and so with the user) through the HMAC signature.

The values of the low-level chain are shorter than the hash function outputs. By truncating the hash output, the estimated collision resistance of the algorithm is also reduced, in this case to 64 bits. NIST recommendation for applications using HMACs [21] is that a security strength between 64 to 96 bits is sufficient for the most of them, and even shorter lengths may be satisfactory if a minimum rate of collisions is not critical for the system. In the proposed protocol, low-level chains elements are delivered fast and have a very short lifetime, setting the time for a computational attack to a cents of milliseconds. Moreover, a tiny false acceptance rate is not critical since the sensing of a channel is repeated multiple times and the final decision depends on a group of different nodes. Thus, a length of 64 bits is secure enough for the requirements of the application.

Nevertheless the chain is not vulnerable to pre-computation attacks since we use a salt to create all the values of the chain. The salt effectively makes it unfeasible to compute a table of all input-output pairs of the generation function, since the input of the function is not 64 bits long but around twice this size.

3.3. Security Justification

Providing authenticity to the sensing reports of a cooperative CRN is the first step towards building a robust sensing mechanism capable of getting vacant channels and primary user detection rates, even in the presence of sensing failures and malicious users.

When reports cannot be authenticated, the fusion centre must merge the data of different users using a non-memory system. For example, if the Majority rule is employed, the system can support up to $50\,\%$ of erroneous inputs. However, if the fusion centre can track the behavior of the users, this information can be used to learn which are the most confident reports and thus, obtain better results [7]. For example, tests based on the likelihood ratio test (LRT) and with a memory of 50 sensings, can support up to $75\,\%$ of erroneous inputs from users that occasionally fall in a dark area where they cannot sense the environment well. If the system knows the attacker profile, better algorithms can be designed to prevent them from altering the sensing decision.

4. Conclusions

In this paper, we have identified the security vulnerabilities of a cooperative sensing process and its prejudicial effects in CR networks. We have proposed a secure protocol for centralized based systems that essentially uses symmetric signatures and one-way chains. The protocol enables the fusion centre to verify the identity of network members and to ensure the received sensing information is really originated from the claimed source. One of the main features of the proposal is the fact that is computationally efficient and introduces a very small bandwidth overhead.

The most demanding phase of the protocol is the registry one, when public key operations must be performed. Anyway, this phase has to be executed only once and for this reason it does not suppose a problem, not even for mobile users that operate in different CRNs. Note that the main applications for CRNs are broadband Internet for rural areas, and specialized applications (i.e. hot-spots, medical apps) for urban ones. The first case of applications must support mobile users that travel by car or train. Rural CRNs are implemented using the IEEE 802.22 standard that provides a communication range around 17-30 km (it can reach 100km depending on EIRP). The diameter of the networks is large enough to allow users that move and leave a particular CRN to enter in a new one, can compute the registry in the new CRN using public/private key methods effectively, without noticing delays.

Urban CRNs cells are smaller in size. However, users use to be more static (they are on a fixed point or moving slowly).

Applications requiring the mobility of users between different cells are usually managed by a central entity. Then, the session of a user in the old cell can be transferred to the new one (handoff) when he travels from one area to the other and so, the overhead of the network is small.

Referencias

[1] Federal Communications Commission, Spectrum policy task force report, Tech. rep., ET Docket No. 02-135 (2002).

[2] J. Mitola III, G. Maguire Jr, Cognitive radio: making software radios more personal, IEEE personal communications 6 (4) (1999) 13–18.

[3] S. Mishra, A. Sahai, R. Brodersen, Cooperative sensing among cognitive radios, in: IEEE International Conference on Communications, IEEE Computer Society, 2006, pp. 1658–1663.

[4] S. Zarrin, T. Lim, Belief Propagation on Factor Graphs for Cooperative Spectrum Sensing in Cognitive Radio, in: IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN), IEEE Computer Society, 2008, pp. 1–9.

[5] W. Wang, W. Zou, Z. Zhou, Y. Ye, Detection Fusion by Hierarchy Rule for Cognitive Radio, in: Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom), IEEE Computer Society, 2008, pp. 1–5.

[6] R. Chen, J.-M. Park, Y. Hou, J. Reed, Toward secure distributed spectrum sensing in cognitive radio networks, Communications Magazine, IEEE 46 (4) (2008) 50–55. doi:10.1109/MCOM.2008.4481340.

[7] M. Jiménez Blasco, J. Mut Rojas, H. Rifà-Pous, Detección robusta por grupos de señales primarias en redes de radio cognitiva, in: XI Reunión Española sobre Criptología y Seguridad de la Información (RECSI), 2010, pp. 371–376.

[8] P. Kaligineedi, M. Khabbazian, V. Bhargava, Secure cooperative sensing techniques for cognitive radio systems, in: IEEE International Conference on Communications (ICC), 2008, pp. 3406–3410. doi:10.1109/ICC.2008.640.

[9] G. Jakimoski, K. P. Subbalakshmi, Towards secure spectrum decision, in: IEEE International Conference on Communications, IEEE Press, Piscataway, NJ, USA, 2009, pp. 2759–2763.
URL http://portal.acm.org/citation.cfm?id=1817770.1817783

[10] S. D. Ersöz, S. Bayhan, F. Alagöz, Secure spectrum sensing and decision in cognitive radio networks, in: A. Özcan, N. Chaki, D. Nagamalai (Eds.), Recent Trends in Wireless and Mobile Networks, Vol. 84 of Communications in Computer and Information Science, Springer Berlin Heidelberg, 2010, pp. 99–111, doi:10.1007/978-3-642-14171-3_9.

[11] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, M. Galloway, A survey of key management schemes in wireless sensor networks, Computer Communications 30 (11-12) (2007) 2314 – 2341, special issue on security on wireless ad hoc and sensor networks. doi:DOI: 10.1016/j.comcom.2007.04.009.
URL http://www.sciencedirect.com/science/article/B6TYP-4NPG0DB-1/2/82b9e495fc4a8cb20ea1f9a5a57124a7

[12] C. Cordeiro, K. Challapali, D. Birru, N. Sai Shankar, IEEE 802.22: the first worldwide wireless standard based on cognitive radios, in: IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN), IEEE Computer Society, 2005, pp. 328–337.

[13] M. Buddhikot, P. Kolody, S. Miller, K. Ryan, J. Evans, DIMSUMNet: New Directions in Wireless Networking Using Coordinated Dynamic Spectrum Access, in: Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM), IEEE Computer Society, 2005, pp. 78–85.

[14] H. Rifà-Pous, J. Herrera-Joancomartí, Cryptographic energy costs are assumable in ad hoc networks, IEICE Transactions on Information and Systems E92.D (5) (2009) 1194–1196.

[15] L. Lamport, Password authentication with insecure communication, Commun. ACM 24 (11) (1981) 770–772. doi:10.1145/358790.358797.

[16] Y. Sella, On the computation-storage trade-offs of hash chain traversal, in: Financial Cryptography, Vol. 2742 of LNCS, 2003, pp. 270–285.

[17] M. Fischlin, Fast verification of hash chains, in: The Cryptographers' Track at the RSA Conference (CT-RSA), Vol. 2964 of LNCS, 2004, pp. 339–352.

[18] C. Library, Crypto++ 5.6.0 benchmarks, Website, http://www.cryptopp.com/benchmarks.html (2009).

[19] H. Rifà-Pous, J. Herrera-Joancomartí, Computational and energy costs of cryptographic algorithms on handheld devices, Future Internet 3 (1) (2011) 31–48. doi:10.3390/fi3010031.
URL http://www.mdpi.com/1999-5903/3/1/31/

[20] D. Lee, Hash function vulnerability index and hash chain attacks, in: IEEE Workshop on Secure Network Protocols (NPSec), 2007, pp. 1–6.

[21] Q. Dang, Recommendation for applications using approved hash algorithms, in: U. D. of Commerce (Ed.), Computer Security, NIST Special Publication, 800:107, 2009.