Towards Secure Spectrum Decision

G. Jakimoski and K. P. Subbalakshmi

Abstract— **The key idea of dynamic spectrum access (DSA) networks is to allow the secondary, unlicensed users to detect and use unused portions of the spectrum (white spaces) opportunistically. The two main constraints in the design of DSA networks is to make sure that this opportunistic access is done without any disruption of service to the primary users and without any modifications to the primaries themselves. Most architectures and protocols for DSA networks in the literature assume that all parties are honest and that there are no attackers. Recently (IEEE ICC, CogNet 2008) we demonstrated the failure of this approach by showing that an attacker can manipulate messages to convince the parties involved in the protocol to make incorrect spectrum decisions. In this paper, we consider spectrum decision protocols in clustered infrastructure-based dynamic spectrum access networks where the spectrum decision in each cluster is coordinated by some central authority. We propose an efficient and provably secure protocol that can be used to protect the spectrum decision process against a malicious adversary.**

Keywords— **dynamic spectrum access, cognitive radio, security, spectrum decision**

I. Introduction

Traditionally, the spectrum has been regulated by governmental agencies and the spectrum bands are assigned to license holders or services on a long term basis for large geographical regions. These fixed spectrum assignment policies have led to under-utilization of the available spectrum. The inefficiency in the spectrum usage and the limited available spectrum for wireless applications gave rise to Dynamic Spectrum Access (DSA) as a new communication paradigm [1]. The DSA networks use the idle licensed spectrum bands for communication, and they vacate the licensed bands upon the return of the primary licensed users. Cognitive radio [2], [3], [4] technologies provide means for making wireless systems computationally intelligent. Hence, cognitive radio technologies has emerged as a natural platform for implementation of DSA networks since they provide tools for opportunistic spectrum usage in an intelligent way.

A plethora of protocols and techniques for DSA implementation have been proposed so far (e.g., [5–10]). Minimal disruption of the primary licensed users and efficient utilization of the idle spectrum bands are two common design goals of the proposed protocols and techniques. In [11], we point out that most of the existing proposals achieve these goals assuming that there are no network attackers and the participants involved in the protocols are honest. If a malicious adversary is introduced into the model, then the existing proposals will fail to fulfill the main objectives of minimal disruption of the primary users and efficient utilization of the vacant spectrum bands.

The main benefit of introducing security in the spectrum decision process is a stronger guarantee that the service of the primary users will not be significantly disrupted. At no additional cost, the resilience of the spectrum decision to malicious attackers protects the secondary network as well. For instance, the DoS types of attacks presented in [11] will not be applicable if the spectrum decision is secured.

One might approach the design of a secure spectrum decision protocol by first designing a spectrum decision protocol, and then applying the existing security technologies to make it secure. However, we note that such approach may fail or lead to less efficient solution than a solution that takes security into consideration from the start. For instance, the information about the channel availability in [17]is not conveyed in a message, but by using a boosting technique to provide greater efficiency. However, the cryptographic schemes that provide data authenticity such as digital signatures and message authentication schemes operate on messages. So, one cannot use the standard cryptographic schemes to provide authenticity of the data exchanged among the parties in the protocol. In [11], we show how a malicious outsider can easily mount a DoS attack on the channel evacuation protocol proposed in [18] by sending a fraudulent warning. One can avoid this by signing the warning messages. However, the protocol will still have reliability/security weaknesses due to the fact that the channel evacuation decision is not made in a distributed fashion. Namely, if a single faulty/corrupt network node sends a warning message, then all network nodes will evacuate the channel although it is available.

In this paper, we propose a protocol designed to provide secure spectrum decisions in a clustered infrastructure-based network where the spectrum decisions are made periodically and independently in each cluster. The proposed protocol guarantees that a malicious outsider and a limited number of corrupt insiders (i.e., nodes that participate in the protocol) cannot have a significant impact on the spectrum decision. The protocol is provably secure, and it is more efficient than the straightforward solutions involving digital signatures or key establishment protocols.

The paper is organized as follows. A description of the network model used in our analysis is given in Section II. The secure spectrum decision protocol is presented in Section III. The security of the protocol against forgers and Byzantine faults is analyzed in Section IV. In Section IV-C, we briefly discuss a somewhat more memory efficient variant of the protocol. The paper ends with concluding remarks.

The authors are with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA, e-mail: goce.jakimoski@stevens.edu, ksubbala@stevens.edu.

This work was funded in part by NSF CT grant number: 0627688 and US Army ARDEC/Picatinny Arsenal.

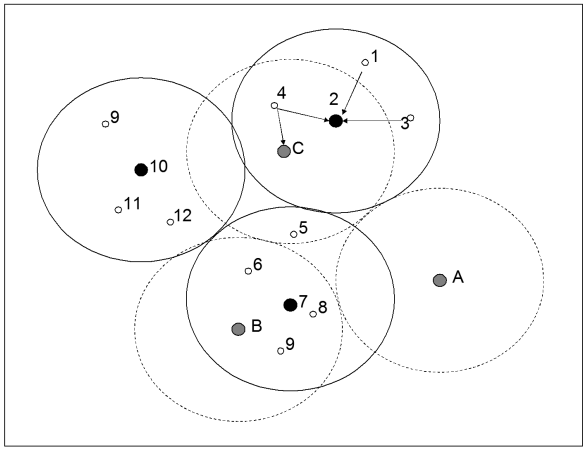

Fig. 1. Two secondary networks coexisting in the same region. The cluster heads of the second network may use the spectrum sensing data obtained from the nodes of the first network.

II. The Network Model

We consider a clustered infrastructure-based dynamic spectrum access network. Each cluster consists of a cluster head and mobile nodes that are within the range of the cluster head. The cluster head has the role of a central authority that exerts some control over the other nodes in the cluster. A base station or an access point are examples of such authorities. We assume that there is an infrastructure that connects the different cluster heads, and this infrastructure is used to establish communication between any two mobile nodes in the network.

The channel availability is decided by the cluster head based on the sensing data provided by the nodes in the cluster. The cluster nodes sense the spectrum periodically, and send their resultant data to the cluster head. The cluster head uses the received information to determine which channels are not used by the primary users, and then it notifies the cluster nodes about the communication channels they should use for uplink and downlink communication. In general, we assume that more than one secondary network exists in a given geographical region. So, we allow, but not require, the spectrum measurements of a given node to be used by more than one network. Figure 1 depicts two secondary networks that coexist in a given region. Nodes 1, 2, 3 and 4 belong to a same cluster of the first network. The nodes 1, 3 and 4 periodically sense the spectrum to detect white spaces or return of the primary, and broadcast their measurement. The cluster head (node 2) uses this information to decide which channels are available. Node C is a cluster head in the second network. Although the node 4 does not belong to the second network, it is within the range of C, and in general, C can use the measurements broadcast by node 4 to determine whether certain channels are available or not.

To provide authenticity of the data exchanged in the spectrum sensing protocol, the nodes will need keys to perform some cryptographic operations. We assume that the keys are distributed using a certificate-based key distribution scheme. That is, some certification authority inducts the nodes into the network by issuing certificates to the nodes. Every node, including the cluster heads, is associated with a unique identifier (ID) and a description of its credentials. For instance, a mobile n may be associated with some unique bit string $\mathrm{ID_n}$ and credentials S_n. The credentials of the node may consist of information describing the real-life identity of the owner of the mobile device, duration of validity of the credentials, etc. Each node n generates a secret signing key s_n and a public verifying key v_n of some digital signature scheme (e.g., DSA or RSA). The certificate issued to n by the certification authority may consist of the unique ID of n, the public key v_n, some additional information (e.g., a timestamp, duration of the certificate, etc.) and a digital signature of the certification over the previous data. To check the validity of the public key v_n of a given node n, the other nodes verify the validity of the certificate using the public key of the certification authority.

III. Description of the protocol

We assume that there is some sort of synchronization among the nodes in the cluster. The time is divided into equal length intervals (or cycles). The nodes know when each cycle begins and ends, and they are also aware of the schedule of the events during a cycle (e.g., which node sends its channel availability data, which channels it uses, etc.). There are three main events that are handled in a given cycle: one or more nodes may join the spectrum decision process in a given cluster, the nodes of the cluster send their spectrum sensing data, and the cluster head sends to the other nodes the final channel assignment. In the following, we briefly describe how each of this operations can be accomplished in a secure and efficient manner. We do not deal with the details of the data sent by the nodes during the spectrum decision. We simply present techniques that enable secure transmission.

A. Join operation

To join the spectrum deciding procedure, a node first generates a sequence of (symmetric) keys $K_0, \ldots, K_n$. The sequence of keys is generated using iterative application of a hash function F to some initial value. Let us denote v consecutive applications of the function F as $F^v(x) = F^{v-1}(F(x))$, and let $F^0(x) = x$. The node has to pick randomly some initial key value K_n and to pre-compute n key values $K_0, \ldots, K_{n-1}$, where $K_i = F^{n-i}(K_n), i = 0, \ldots, n$. The sequence of key values is called a key chain. Suppose that the join operation is performed in the t_s^{th} cycle. Then, the node sends a message $\langle t_s, \mathrm{ID_H}, \mathrm{ID_N}, \mathrm{K_0}, \mathrm{D}, \mathrm{sign_N(t_s, ID_H, ID_N, K_0, D)}\rangle$ consisting of the current cycle number t_s, unique ID of the node N, the key K_0, some additional data D that is used for purposes irrelevant to our discussion, and a digital signature $sign_N(t_s, \mathrm{ID_H}, \mathrm{ID_N}, \mathrm{K_0}, \mathrm{D})$. The cluster head H checks the authenticity of the message using the pub-

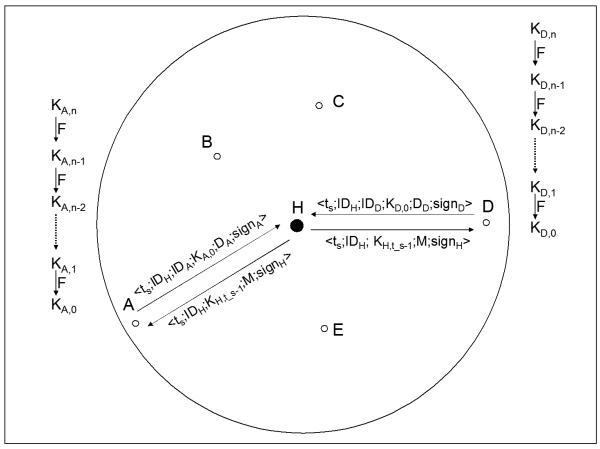

Fig. 2. The nodes A and D join the sensing protocol. Each node generates a chain of keys and sends the first key of the chain K_0 to the cluster head. The key K_0 serves as a commitment to the rest of the keys in the chain. The cluster head sends back communication instructions and a key K_{H,t_s-1}. The key K_{H,t_s-1} serves as a commitment to the keys that are used by the cluster head in the subsequent cycles.

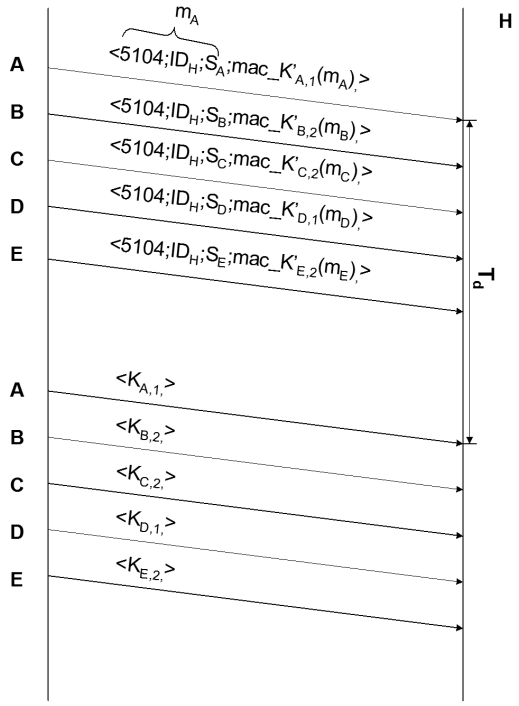

Fig. 3. An example of sensing data transmission process. The cluster is the one depicted in Figure 2. We assume that the current interval is 5103, A and D joined in interval 5102, and B, C and E joined in interval 5101. To verify the authenticity of the sensing data S_C sent by C, the cluster head checks whether $K_{C,1} = F(K_{C,2})$. If so, the cluster head computes $K'_{C,2}$ and checks the validity using the MAC of the message.

lic key of N. If the verification is successful, the cluster head stores the identity of N and the related information in a table. The cluster head sends back a signed message $\langle t_s, \mathrm{ID_H}, \mathrm{K_{H,t_s-1}}, \mathrm{M}, \mathrm{sign_H}(\mathrm{ID_H}, \mathrm{K_{H,t_s-1}}, \mathrm{M})\rangle$ that includes the unique ID of the cluster head, the symmetric key K_{H,t_s-1} used by the cluster head in the previous interval $t_s - 1$ and some information M. The information M includes communication parameters for the nodes that have joined in the current interval. For instance, it may contain time and frequency schedule for submitting sensing data, available channels for ordinary communication, etc. The messages exchanged during the join operation are depicted in Figure 2.

B. Sending the channel availability data

Each key $K_i, i > 0$, of the key chain generated when the node N joined the spectrum decision procedure is used in the cycle $t_s + i$ to protect the authenticity of the sensing related information broadcast by N. At the beginning of each cycle, the nodes sense the spectrum to detect white spaces. Each node sends the resulting information to the cluster head as follows. A given node N constructs a message $m = \langle t_s+i, \mathrm{ID_H}, \mathrm{ID_N}, \mathrm{S}\rangle$, where t_s+i is the current interval, t_s is the interval when N joined, and S is the sensing related information. Then N broadcasts $\langle m, mac_{K'_i}(m)\rangle$, where the authentication tag $mac_{K'_i}(m)$ is a MAC of m computed using a symmetric key K'_i derived from K_i. After some time delay T_d, the node N reveals the key K_i by broadcasting it. The cluster head verifies whether the received value K_i is valid by checking whether $F(K_i)$ is equal to K_{i-1}. If so, it derives a key K'_i and checks the authenticity of the received message. If the message is valid, then the cluster uses the received data S in the spectrum decision. An example is given in Figure 3.

C. Broadcasting the spectrum decision

Once the cluster head successfully verifies the messages sent by the nodes, it uses the information to decide which channels are available, and notifies the nodes of the channel assignment as follows. The cluster head constructs a message $m = \langle t, \mathrm{ID_H}, \mathrm{I}\rangle$, where t is the current interval, $\mathrm{ID_H}$ is the unique ID of the cluster head, and I may consist of channel availability, channel assignment and/or other information. Then, the cluster head broadcasts $\langle m, mac_{K'_t}(m), sign_H(m)\rangle$, where $mac_{K'_t}(m)$ is an authentication tag computed using the key K'_t derived from the key K_t of the H's key chain, and $sign_H(m)$ is a digital signature of m. After some delay, the cluster head reveals K_t. The nodes that have been in the cluster in the previous intervals check the validity of K_t by verifying that $F(K_t) = K_{t-1}$. If so, they use K_t to check the authenticity of I. The nodes that are new to the cluster check the validity by using a more expensive digital signature verifying algorithm.

Once the spectrum decision protocol is over, the nodes

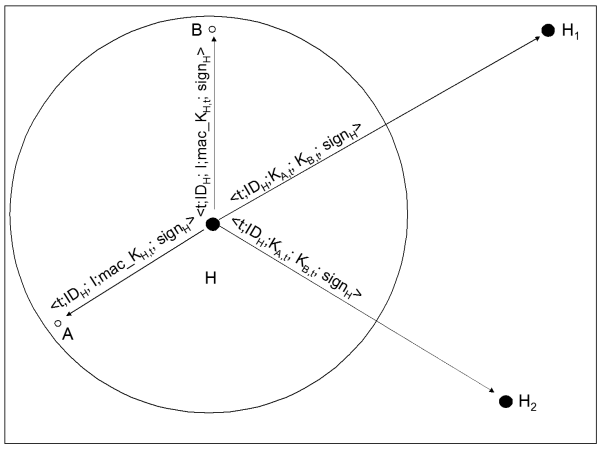

Fig. 4. Based on the spectrum sensing data received from the nodes in the cluster, the cluster head makes a spectrum decision and sends it back to the nodes in the cluster. The old nodes of the cluster use a symmetric key to verify the authenticity of the message. The new nodes verify the authenticity of the message using the digital signature. The cluster head also sends the keys that the nodes have revealed to its neighboring cluster heads.

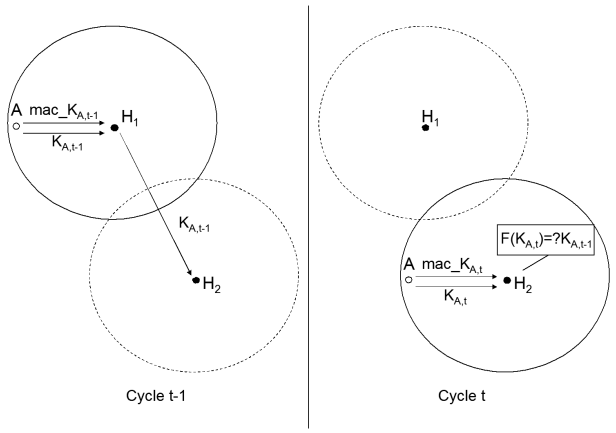

Fig. 5. The node A has crossed into a new cluster. The cluster head H_2 has received the key $K_{A,t-1}$, and can verify the validity of the key $K_{A,t}$ used by A to protect the integrity of its sensing data by checking whether $F(K_{A,t})$ is equal to $K_{A,t-1}$.

resume normal communication using the available channels. The cluster head performs one last step. It securely sends the keys used by the nodes in the given interval to the neighboring cluster heads. The cluster head also receives the keys used by the nodes in the neighboring clusters. That way the mobile nodes will not have to perform the join operation each time they cross into a different cluster. For instance, suppose the node N moved to a neighboring cluster in the interval $t_s + i$. The new cluster head has received the key K_{i-1} in the previous interval, and it can use K_{i-1} as a commitment to the key K_i that N will use in the interval $t_s + i$. So, the nodes will only need to perform the expensive join operation only when they are new to the network or run out of keys in their key chain. The messages sent during the spectrum decision broadcast are depicted in Figure 4, and the crossing of a node into a new cluster is depicted in Figure 5.

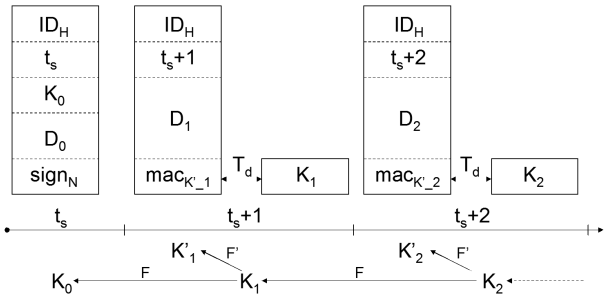

Fig. 6. The transcript τ_N of node N and the generation of the key chain.

IV. Security and performance analysis

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

A. Data authenticity and resilience to Byzantine faults

To show that our protocol provides authenticity of the messages sent by the nodes we define the *transcript* τ_N of a node N to consist of the sequence of messages sent by N ordered by the time of their transmission. The transcript τ_N of the node N and the computation of keys and authentication tags are depicted in Figure 6. The security-irrelevant data of the messages sent by N is denoted by D_i. Note that the pair $\langle ID_H, t\rangle$ is a nonce (i.e., does not repeat) since the nodes send only one message per interval and the intervals are uniquely numbered within the clusters. While the nodes send one message per cycle, the cluster head sends more than one message. We assume that the messages sent by the cluster head contain a type field indicating the type of the different messages sent by the cluster head. The triple consisting of the ID of the head, the interval number and type is (should be) unique for each message.

The unforgeability of the transcripts follows from the following theorem. Due to space limitation, we state the theorem informally and omit its prove.

Theorem 1: Suppose that:

1. the digital signature scheme, which is used to bootstrap the scheme, is unforgeable,
2. the function $F(K)$ is collision resistant and is computed as $F(K) = f_K(0)$, where f is a pseudorandom function,

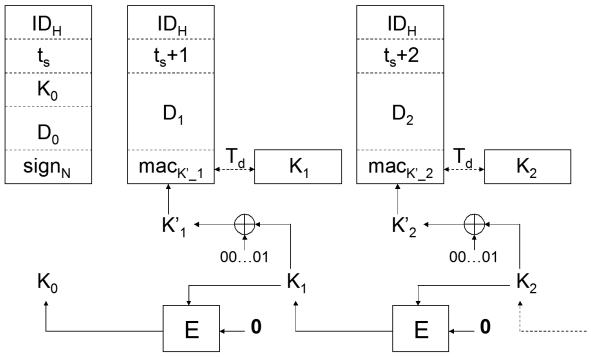

Fig. 7. An implementation using a block cipher resistant to related-key cryptanalysis

3. the MAC scheme, which is used to authenticate the chunks of the stream, is unforgeable even if the adversary has access to the commitment $F(K)$ to the secret key K used by the MAC scheme, and
4. F' is an identity mapping.

Then, a malicious adversary cannot forge a transcript of an honest node.

We suggest to implement the function F using a block cipher as in [13]. The key value $K_{i-1} = E_{K_i}(0)$ is derived by encrypting zero using the key K_i. The keys K'_i that are used to compute the authentication tags are derived by flipping the value of the last bit of the corresponding key K_i (i.e., $K'_i = K_i \oplus 0^{n-1}1$). The computation of the authentication tags for a given transcript is depicted in Figure 7. The unforgeability of the transcripts is based on the assumption that $F(x) = E_x(0)$ is collision resistant and the block cipher E in use is a pseudorandom permutation and it is resistant to related-key attacks [12], [14].

The authenticity of the messages exchanged by the nodes during the spectrum decision does not guarantee that the spectrum decision will be correct. It is possible that the adversary can corrupt nodes and use the corrupt nodes to "trick" the honest nodes into making a wrong spectrum decision. So, we have to analyze the resilience of the protocol to Byzantine faults [15], [16]. Byzantine faults are used in reliability analysis of distributed systems to model failures without making specific assumptions about the behavior of a faulty process. It is assumed that the faulty process can act maliciously (i.e., send messages when it is not supposed to, make conflicting claims to other processes, act dead for awhile and then revive itself, etc). To analyze the resilience of the proposed protocol to Byzantine faults, one must make more specific assumptions about the spectrum decision process. Assuming that the cluster head is non-faulty (honest) and the spectrum decision is reached using a majority voting, we have that the spectrum decision is resistant to up to $\lceil N_m/2 \rceil - 1$ Byzantine faults, where N_m is the number of nodes in the cluster. Some other trust-based schemes that give different weights to the data sent by different nodes can be employed as well. However, proposing and comparing such schemes is out of the scope of this paper.

B. Advantages over some straightforward approaches

One straightforward solution that provides authenticity of the messages exchanged by the cluster nodes is to digitally sign the messages. However, due to the large number of messages, digitally signing each message will introduce a relatively large computational overhead. In our case, the mobile nodes use digital signatures only when they join the spectrum decision protocol, and the cost of computing a digital signature is amortized over many intervals.

Another possible solution is the nodes to establish keys with the cluster heads. This key is then used in a message authentication scheme to protect the integrity of the data sent by the cluster node to the cluster head. There are two possible variants. In the first variant, the mobile nodes engage in a key establishment protocol whenever they cross into a new cluster. If the topology of the network is very dynamic due to node mobility, then this solution might not be very efficient. It would more efficient if the node engages in a key establishment only when joining the spectrum decision protocol. The neighboring cluster heads inform each other about the keys used by their nodes. So, when a cluster node moves to a new cluster, it can continue using its old key.

The main advantage of the last variant is that the protocol is somewhat more efficient than our proposal. The cluster nodes compute and append authentication tags to the messages they send. The cluster head will check the authenticity of the messages by checking the validity of the authentication tags. In our protocol, besides computing the authentication tags, the cluster nodes have to generate a new key per message. In addition, the cluster head will have to verify the validity of the key before checking the validity of the authentication tags. However, both the cost of key generation and the cost of key validation are negligible in our proposed implementation (approximately one block encryption).

On the other hand, there are two advantages of our protocol. The first advantage is that it allows cluster heads to verify authenticity of spectrum measurement data sent by cluster nodes that belong to other networks. The second advantage is that the cluster heads do not have to provide secrecy of the commitments sent to other cluster heads. Let us recall that when a node moves to a neighboring cluster, the cluster head of the new cluster uses the commitment received in the previous interval(s) to verify the authenticity of the messages sent by the node. These commitments are public. If the nodes establish a key with the cluster head, then this key must be kept secret when the cluster head sends it to its neighbors. The knowledge of the key will allow the adversary to forge messages. Furthermore, the disclosure of the key by a corrupt cluster head will allow the adversary to forge messages when the

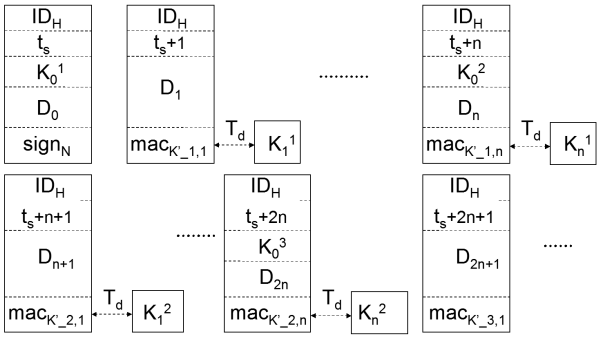

Fig. 8. Instead of being sent in a signed message, the commitment K_0^i of the i-th ($i > 1$) key chain is sent in the message authenticated with the last key $K_n^{(i-1)}$ of the previous key chain.

node moves to other cluster. A corrupt cluster head can help the adversary to produce forgeries in our protocol as well. For instance, it can send to its neighbors a commitment "made" by the adversary as a commitment "made" by an honest node. The adversary can now send messages to the neighboring cluster heads on behalf of the honest node. However, in our protocol, one can easily track down the corrupt node that distributes false commitments. This fact can discourage the corrupt nodes to send false commitments.

C. A linked key-chains variant

The length of the key chain generated when the nodes join the spectrum decision is an important trade-off parameter when it comes to the efficiency of the proposed protocol. If the length of the key chain is large, then the digital signature of the join operation is amortized over a large number of intervals. However, long key chains lead to larger:

- memory requirements due to the increased size of the memory required to store the keys of the chain, and
- start-up delay due to the increased time needed to generate the keys of the chain.

A somewhat more efficient variant is depicted in Figure 8. When joining the network, the node generates a short key. When all keys in the chain are used, the node generates a new short key chain. However, instead of performing the join operation, the node sends the commitment K_0^2 in the last message authenticated using the first chain. This procedure is subsequently repeated. The digital signature is again amortized over a large number of intervals. However, the node will not have to generate and store long key chains.

V. Conclusion

Many existing dynamic spectrum access protocols make spectrum decisions assuming that all parties involved in the spectrum decision are honest and there is no malicious outsider that can manipulate the spectrum decision process. A solution that is secure against malicious adversaries would give a stronger guarantee of minimal disruption to the primary users and would improve the reliability of the secondary network. We have considered spectrum decision in an infrastructure-based DSA network and proposed techniques that provide security against a malicious adversary that cooperates with a limited number of corrupt insiders.

References

[1] FCC ET Docket No 03-322: Notice of proposed rule making, 2003.
[2] J. Mitola III, "Cognitive radio: an integrated agent architecture for software defined radio," Ph.D Thesis, KTH Royal Institute of Technology, 2000.
[3] S. Haykin, "Cognitive radio: brain-empowered wireless communications," IEEE Journal on Selected Areas in Communications, vol. 23, no. 2, pp. 201–220, 2005.
[4] I. F. Akyildiz, W-Y. Lee, M. C. Vuran and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," Computer Networks, vol. 50, pp. 2127-2159, 2006.
[5] I.F. Akyildiz and Y. Li, "OCRA: OFDM-based cognitive radio networks," Broadband and Wireless Networking Laboratory Technical Report, 2006.
[6] R.W. Brodersen, A. Wolisz, D. Cabric, S.M. Mishra and D. Willkomm, "Corvus: a cognitive radio approach for usage of virtual unlicensed spectrum," Berkeley Wireless Research Center (BWRC) White paper, 2004.
[7] M.M. Buddhikot, P. Kolodzy, S. Miller, K. Ryan and J. Evans, "DIMSUMNet: new directions in wireless networking using coordinated dynamic spectrum access," IEEE WoWMoM 2005, pp. 78–85, 2005.
[8] C. Cordeiro, K. Challapali, D. Birru and S. Shankar, "IEEE 802.22: the first worldwide wireless standard based on cognitive radios," IEEE DySPAN 2005, pp. 328–337, 2005.
[9] C. Cordeiro and K. Challapali, "C-MAC: A Cognitive MAC Protocol for Multi-Channel Wireless Networks," IEEE DySPAN 2007, pp. 147–158, 2007.
[10] X. Jing and D. Raychaudhuri, "Spectrum co-existence of IEEE 802.11b and 802.16a networks using CSCC etiquette protocol," IEEE DySPAN 2005, pp. 243–250, 2005.
[11] G. Jakimoski and K. P. Subbalakshmi, "Denial-of-Service Attacks on Dynamic Spectrum Access Networks," In the Proceedings of IEEE CogNet 2008, Beijing, May 19–23, 2008.
[12] E. Biham, "New Types of Cryptanalytic Attacks Using Related Keys," Journal of Cryptology, v.7, n.4, 1994, pp. 229-246.
[13] G.Jakimoski, "Some Notes on the Security of the Timed Efficient Stream Loss-tolerant Authentication Scheme," In the Proceedings of Selected Areas of Cryptography 2006, Lecture Notes in Computer Science, vol. 4356, pp. 345–361.
[14] G. Jakimoski and Y. Desmedt, "Related-key Differential Cryptanalysis of 192-bit Key AES Variants," Proceedings of the 10th Workshop on Selected Areas of Cryptography, LNCS 3006, pp. 208–221, Springer, 2004.
[15] M. Pease, R. Shostak and L. Lamport,"Reaching agreement in the presence of faults," Journal of the ACM, vol. 27, no. 2, pp. 228–234, 1980
[16] D. Dolev and H. R. Strong, "Authenticated algorithms for Byzantine agreement," SIAM Journal on Computing, vol. 12, no. 4, pp. 656–666, 1983.
[17] T.A. Weiss and F.K. Jondral, "Spectrum pooling: an innovative strategy for the enhancement of spectrum efficiency," IEEE Radio Communication Magazine, vol. 42, pp. 8–14, 2004.
[18] X. Liu and Z. Ding, ESCAPE: A Channel Evacuation Protocol for Spectrum-Agile Networks," IEEE DySPAN 2007, pp. 292–303, 2007.