VTE-AKA: an optimum vector time encoding scheme for synchronising authenticated key agreements in cognitive radio ad hoc networks

Sanjay Kumar Dhurandher¹, Gaurav Indra²
¹Department of Information Technology, Netaji Subhas University of Technology, New Delhi, India
²Division of Information Technology, Netaji Subhas Institute of Technology, University of Delhi, New Delhi, India
E-mail: dhurandher@gmail.com

Abstract: In a hostile wireless environment, a cognitive radio ad hoc network, a robust-spectrum aware paradigm, is susceptible to spectrum sensing data falsification at the data link layer. In the state-of-the-art literature, distributed cooperative spectrum sensing is considered as a competent technique to ameliorate the performance of primary users (PUs) and malicious users (MUs) detections but synchronising local clocks at secondary users (SUs) remains a major security concern. In this study, an optimal vector time encoding technique infused with an optimised authenticated key agreement scheme, termed as VTE-AKA is proposed for logically synchronising localised clocks at authenticated SUs. VTE-AKA competently offers the security attributes viz. key compromise impersonation resilience, known session key security, full perfect forward secrecy, key replicating resilience, unknown key-share resilience, replay resilience, reflection resilience, and key control resilience. VTE-AKA is simulated and validated in a cognitive scenario where results validate the adeptness of VTE-AKA in PU's and MU's detection, average network throughput, and overall channel utilisation.

1 Introduction
A cognitive radio ad hoc network (CRAHN), an intelligent, autonomous, and spectrum aware communication network based on dynamic spectrum access, alleviates spectrum scarcity problem by cognitively sensing its electromagnetic environment and dynamically re-configuring its communication parameters. It facilitates secondary users (SUs) to deviously utilise the sporadically used licenced spectrum bands without interfering with licenced primary user (PU). Spectrum sensing requirements are quite high for a single localised SU. Thus, distributed cooperative spectrum sensing (DCSS) is adopted to deduce individual sensing requirements by fostering either hard or soft cooperation among ad hoc SUs in an untrusted radiofrequency (RF) environment. Besides, DCSS improves PUs and MUs detection probabilities by overcoming hidden node problems, multipath fading, shadowing, receiver uncertainty, and obstacle blocking in an untrusted RF environment. The adversaries exploit open networks either by snooping common control channel (CCC) for a free channel list (FCL) or corrupting an exchanged selected data channel (SDCH) report. The security of the FCL and SDCH report rests on the standard cryptographic primitives and are ensured in synchronised ad hoc SUs. Hence, it becomes imperative to logically synchronise clocks at SUs as physical synchronisation does not offer distinguishability amidst concurrent events. The motivation for VTE-AKA comes from the vital need for an adept scheme for logically synchronising the localised clocks, infused with a computationally optimised light-weight AKA in CRAHNS. The proposed vector time encoding is an extension of the vector time-stamping mechanism from distributed computing and is assimilated for achieving causality and distinguishability in concurrent messages and full resilience against spectrum sensing data falsification (SSDF) in DCSS. The analytic review of the state-of-the-art underlines the necessity of a provably secure and computationally efficient AKA protocol that can proficiently thwart CCC saturation in DCSS and competitively safeguard the session parameters. The vector time encoded-AKA protocol facilitates computation of provably secure unique common session keys for distinct sessions amid SUs and prevents injection of falsified sensing data from the illegitimate SUs claiming false identities. Hence, the VTE-AKA protocol provides the following security attributes:

- Key compromise impersonation resilience (KCI-R)
- Known session key (KSK) security
- Full perfect forward secrecy resilience (fPFS-R)
- Key replicating resilience (KR-R)
- Unknown key-share resilience (UKS-R)
- Replay resilience (R-R)
- Reflection resilience (RF-R)
- Key control resilience (KC-R)

The main contributions in virtue of VTE-AKA are three-fold.

1. The pre-requisite constraint of synchronising distributed local clocks at cognitive SUs is identified as the main hindrance to a secure AKA. Henceforth, a novel vector time encoding in conjunction with an optimised three-pass AKA is proposed.
2. An optimised three-pass AKA is designed for substantially reducing the communication latencies in sessions between cognitive SUs and expediting the computation of unique common session keys.
3. VTE-AKA proficiently reduces computational complexity to a significant level preserving the security attributes and robustness of an AKA. It adeptly maintains the dynamic equilibrium between computational complexity and the desired security threshold for a mutual authentication scheme.

The organisation of the paper: Section 2 reviews the state-of-the-art. Section 3 discusses the CRAHN system and threat model. Section 4 describes the VTE-AKA protocol. Section 5 analyses VTE-AKA's computation and communication complexity. Section 6 reviews security analysis. Section 7 briefs simulation and performance analysis. Section 8 concludes the paper.
computational time and space complexities are on the higher sides. In [25], a symmetric cryptography-based secure medium Data (SSD) and control messages are multi-cast on CCC but it and isolate MUs by refining CSS' reliability. They do not access control (MAC) protocol is proposed for establishing secure proficiently defend the spectrum decision against SSDF in presence of a trusted third party (TTP) is indispensable. TTP pre-counter node impersonation, dictionary, node replay, node compromise, Man-in-the-middle (MITM), and stolen-verifier attacks, while maintaining perfect-forward-secrecy. The review further analyses the impact of prominent security threats viz. Primary User Emulation Attack (PUEA) and SSDF on SUs' dynamic spectrum access and its mitigation techniques. The proposed security and privacy-preserving protocols in [5-21] proficiently defend the spectrum decision against SSDF in Cooperative Spectrum Sensing (CSS) and minimise its impact. In Cognitive Radio Network (CRNs), incentive-driven protocols are proposed in [8, 10, 22] to combat selfish attacks and preserve privacy. In statistical Bayesian networks, the authors in [5, 9] analysed spectrum tenancy by learning correlations, causations, and isolate MUs by refining CSS' reliability. They do not incorporate SUs' distinctive selfish behaviour and its impact on data fusion in DCSS [18]. In [6], a probability-based attack-aware CSS protocol partially isolated MUs from the set of SUs but is prone to a range of cross-layer attacks. A differential evolution-weighted gain CSS scheme for multi-node sensing and efficient data fusion in distributed CRNs is proposed in [23]. It generates distinctive optimal weights for sensing nodes making sensing data independent of received signal features, hence improves sensing performance in DCSS. However, it overlooks two factors viz. diverse nature of MUs and synchronisation of sensing nodes at the network layer.

An information-theoretic secrecy approach for cooperation in PUs and SUs is proposed in [24]. The desired collaboration is formulated as an optimisation problem and the protocol achieves the threshold secrecy by distributing PUs' transmission power. A moral hazard principal–agent-based joint spectrum sensing and access framework is proposed in [7] to thwart SSDF. It observes the misconduct of MUs and incorporates an incentive mechanism to punish them. Yet, the defence is inept to predict the impact of MUs on CSS [16]. In [13], a scalable CoG-Auth protocol is proposed to counter the self-co-existence in ad hoc CRNs. It keeps a key hierarchy linking the temporal, partial, and session keys and centralised certificate authority is made obsolete. It is robust but computational time and space complexities are on the higher sides. Competent protocols are proposed in [11, 12, 20] for intrusion detection, efficient sensing and CSS defence, proficient data fusion, and spectrum decisions against MUs. Spectrum Sensing Data (SSD) and control messages are multi-cast on CCC but it overlooks MUs’ attack strength and intelligent collusive strategies [15]. In [25], a symmetric cryptography-based secure medium access control (MAC) protocol is proposed for establishing secure communication sessions between cognitive SUs. It achieves secrecy over data transmission channels and CCC by securing the MAC layer. Yet, it is susceptible to crucial security threats viz. key replicating and key compromise impersonation attack [14]. In [17, 22], two-tier Mutual Authentication and Key Agreement (MAKA) and byzantine defence techniques are proposed, which rely on symmetric–asymmetric cryptosystems incorporating identity validation from a single authorised centre and hence claims to be resistant against MITM, Denial-of-Service (DoS), and reflected attacks and without using a digital signature. Nonetheless, it has a central point of failure and is not immune to vital security threats from the state-of-the-art.

2 Literature review

In the CRAHN, AKA in SUs is of prime concern, and hence the presence of a trusted third party (TTP) is indispensable. TTP pre-distributes static and ephemeral public–private keys among SUs. In the state-of-the-art literature, light-weighted elliptic curve cryptography (ECC) with lesser computational complexity and lower communication overhead has replaced the first generation public key cryptosystems viz. Diffie Hellman and RSA. The literature highlights the pros and cons of the state-of-the-art ECC-based mutual authentication, key agreement, and data fusion protocols employed in CRAHNs. ECC-based optimal key pre-distribution and session management schemes are proposed in [1, 2] for maintaining a significant trade-off between node connectivity, resilience, and storage pre-requisites. ECC two-factor user authentication protocols for secure communications in ad hoc wireless sensor networks (WSNs) are proposed in [3, 4], which counter node impersonation, dictionary, node replay, node compromise, Man-in-the-middle (MITM), and stolen-verifier attacks, while maintaining perfect-forward-secrecy. The review further analyses the impact of prominent security threats viz. Primary User Emulation Attack (PUEA) and SSDF on SUs' dynamic spectrum access and its mitigation techniques. The proposed security and privacy-preserving protocols in [5-21] proficiently defend the spectrum decision against SSDF in Cooperative Spectrum Sensing (CSS) and minimise its impact. In Cognitive Radio Network (CRNs), incentive-driven protocols are proposed in [8, 10, 22] to combat selfish attacks and preserve privacy. In statistical Bayesian networks, the authors in [5, 9] analysed spectrum tenancy by learning correlations, causations, and isolate MUs by refining CSS' reliability. They do not incorporate SUs' distinctive selfish behaviour and its impact on data fusion in DCSS [18]. In [6], a probability-based attack-aware CSS protocol partially isolated MUs from the set of SUs but is prone to a range of cross-layer attacks. A differential evolution-weighted gain CSS scheme for multi-node sensing and efficient data fusion in distributed CRNs is proposed in [23]. It generates distinctive optimal weights for sensing nodes making sensing data independent of received signal features, hence improves sensing performance in DCSS. However, it overlooks two factors viz. diverse nature of MUs and synchronisation of sensing nodes at the network layer.

An information-theoretic secrecy approach for cooperation in PUs and SUs is proposed in [24]. The desired collaboration is formulated as an optimisation problem and the protocol achieves the threshold secrecy by distributing PUs' transmission power. A moral hazard principal–agent-based joint spectrum sensing and access framework is proposed in [7] to thwart SSDF. It observes the misconduct of MUs and incorporates an incentive mechanism to punish them. Yet, the defence is inept to predict the impact of MUs on CSS [16]. In [13], a scalable CoG-Auth protocol is proposed to counter the self-co-existence in ad hoc CRNs. It keeps a key hierarchy linking the temporal, partial, and session keys and centralised certificate authority is made obsolete. It is robust but computational time and space complexities are on the higher sides. Competent protocols are proposed in [11, 12, 20] for intrusion detection, efficient sensing and CSS defence, proficient data fusion, and spectrum decisions against MUs. Spectrum Sensing Data (SSD) and control messages are multi-cast on CCC but it overlooks MUs’ attack strength and intelligent collusive strategies [15]. In [25], a symmetric cryptography-based secure medium access control (MAC) protocol is proposed for establishing secure communication sessions between cognitive SUs. It achieves secrecy over data transmission channels and CCC by securing the MAC layer. Yet, it is susceptible to crucial security threats viz. key replicating and key compromise impersonation attack [14]. In [17, 22], two-tier Mutual Authentication and Key Agreement (MAKA) and byzantine defence techniques are proposed, which rely on symmetric–asymmetric cryptosystems incorporating identity validation from a single authorised centre and hence claims to be resistant against MITM, Denial-of-Service (DoS), and reflected attacks and without using a digital signature. Nonetheless, it has a central point of failure and is not immune to vital security threats from the state-of-the-art.

3 Cognitive radio ad-hoc network model

Consider $P = \{1, 2, 3, \ldots, P\}$, set of PUs, $U = \{1, 2, 3, \ldots, N\}$, set of SUs, and $C = \{1, 2, 3, \ldots, K\}$, set of M non-overlapping channels in a licenced spectrum. The set of MUs is defined as $O = \{1, 2, 3, \ldots, n\}$, where $O \cup U$ and $n \neq 0$. SUs validate the presence of PUs on the licenced spectrum by detecting the signal modulation and PUs utilise the allocated licenced spectrum without any prior notice. $SU_i \in U$, the list of idle, non-overlapping channels is maintained by $SU_i$, and is denoted by $I_i = \{1, 2, 3, \ldots, k\}$; $I_i \subseteq C$ and $k \neq 0$.

CRAHN is modelled by a connectivity graph $G(V, E)$, where the set of vertices $V$, models distribution of SUs and set of edges, $E$ models communication links in SUs. Furthermore, an edge between $SU_i$ and $SU_j$, is considered as an established link, only if they are within each others radio range and $I_i \cap I_j = \emptyset$. $SU_i \in U$, the one-hop neighbourhood is defined as the set $\mathcal{N}_i = \{1, 2, 3, \ldots, m\}$.

3.1 System architecture

The CRAHN system model consists of licenced PUs, unlicensed cognitive users (SUs), infiltrated malicious users (MUs) posing as legitimate SUs, and full-duplex cognitive hybrid access points (F-CHAPs) operating on a licenced spectrum band. SUs are segregated into $k$ finite sets using a $k$-medoids spectrum aware clustering scheme. For secure and efficient spectrum access, the sharing of authenticated sensing data in the DCSS paradigm is indispensable and accomplished through Cognitive Radio (CRs) interacting on non-overlapping communication channels. Control messages are exchanged among SUs and F-CHAPs, for spectrum-aware routing, distributed spectrum access, and authentication of SUs. MUs infiltrate the CRAHN system, posing as legitimate SUs, and penetrate the trust model by injecting counterfeit sensing reports in DCSS resulting in SSDF. A finite cluster $c_i$ is enclosed within the cluster heads' radio transmission range as shown in Fig. 1.

3.2 Threat model

In the CRAHN model, an AKA is vital for parties communicating over public channels, in which communications can be entirely controlled by adversaries. In VTE-AKA, two intended parties interchange messages and compute a unique common secret session key as the function of at least four secret information viz. long-term and ephemeral secret keys of both parties. The critical security threat is the compromise of secret session key or long-term and ephemeral secret keys to an eavesdropper.

Under the extended Canetti–Krawczyk security model [13], all possible security threats resulting from compromised session-specific secret information viz. long-term and ephemeral secret keys are captured for achieving a stronger sense of security. Yet, an adversary is permitted to reveal any subset of the four keys (two long-term and two ephemeral secret keys), which does not include both the long-term and ephemeral secrets of the same communicating party, as this would break the scheme. In the extended Canetti–Krawczyk security model, the fundamental security primitive of VTE-AKA is the passive attack resilience against the adversary, who may obtain a unique common secret

© The Institution of Engineering and Technology 2020

IET Commun., 2020, Vol. 14 Iss. 20, pp. 3529-3540

3530
session key or ephemeral secret keys by merely eavesdropping on communication sessions.

Under the threat archetype of an extended Canetti–Krawczyk security model, an active adversary is capable of counterfeiting identities, forging communication of one of the participants during protocol execution, intercepting the session messages, injecting falsified sensing reports, deleting and replacing digital certificates, and replaying the test or past session parameters in any instance of VTE-AKA. Precisely, an active adversary may reveal either both ephemeral keys or both long-term keys or one of each from two different participants, once the session is established. In the passive sessions, an adversary has only eavesdropping ability and no active intervention is permissible in communication session establishments. Nevertheless, an important security threat in a practical scenario is information leakage, as ephemeral secret session key or ephemeral secret keys by merely eavesdropping on spectrum sensing reports, ensuring data confidentiality and integrity.

The security of VTE-AKA is based on the complexity involved in solving the discrete logarithm problem in ECC (ECDDL). In the proposed scheme, E is an elliptic curve defined over a finite field $F_q$ and $P \in E(F_q)$ is a point of prime order $n$.

4.1 Conjectures

4.1.1 Discrete logarithm problems in ECC: ECDLP is intractable, if for any probabilistic polynomial time Turing machine $\Xi$ with knowledge of $R = a \cdot P$ and $S = b \cdot P; R, S \in (P)$, probability of success in determining $\log_P R = x \in [1, n - 1]$ is negligible

$$\exists \in [1, n - 1];\quad R = xP;\quad n = \log_P R$$

4.1.2 Elliptic curve computational Diffie–Hellman problem: ECDHDP is intractable, if for any probabilistic polynomial time Turing machine $\Xi$ with knowledge of $R = a \cdot P$ and $S = b \cdot P, R, S \in (P)$, probability of success in computing $a \cdot b \cdot P$, is negligible

$$\exists \in [1, n - 1];\quad R = aP;\quad S = bP;\quad n = \log_P R$$

4.1.3 Elliptic curve decisional Diffie–Hellman problem (ECDHP): ECDHP is intractable, if for any probabilistic polynomial time Turing machine $\Xi$ with knowledge of $R = a \cdot P$ and $S = b \cdot P$, probability of success in distinguishing the two probability distributions $(P, R, S)$ and $(P, R, S)$ is negligible

$$\exists \in [1, n - 1];\quad R = aP; S = bP;\quad n = \log_P R$$

4.2 Notations adopted

The notations are listed in Table 1.

4.3 Mutual authentication and key agreement phase

The steps performed by the participating cognitive SUs in a communication handshake are as follows (Tables 2 and 3):

Step 1: Initially, the SU_A arbitrarily selects a random number, $e_A \in [1, n - 1]$ and computes the parameter $E_A$, which acts as a long-term public key and sends it to SU_B

$$E_A = H(\epsilon_A || S_A) \times G$$

Step 2: As SU_B obtains $E_A$ from SU_A, it performs the following:

(i) Using an embedded key validation, SU_B verifies, if $E_A \in (G \ast )$. (2)

(ii) If false, SU_B ends execution and returns 'FAILURE'.

(iii) If true, selects a random number $e_B \in [1, n - 1], where$

$$E_B = H(\epsilon_B || S_B) \times G$$

(iv) Computes

$$l = h \times H(\epsilon_B || S_B) \times S_A \times E_A$$

(v) Crypts the current time $T_c$ and invoke the following functions:

$$\exists (T_C, T_B, \Delta T, \Delta \tau) \in E(A, A) \mid E_B \parallel Z_B \parallel Z_B$$

(v) Sends $m_B$ to SU_A and destroys $e_B$ and $g_B$.

Step 3: Upon receiving $m_B$, SU_A performs the following:

(i) Read arrival time $T_R$.

(ii) Decrypt message $m_B$ and generates message $m_B$

$$m_B \leftarrow D_B(A, A)[m_B]$$

(iii) Verify if $E_B \in (G \ast )$ and invoke the function:

$$\exists (T_C, T_B, T_B, \Delta T, \Delta \tau)$$

(iv) Now generate the below-mentioned parameters:

$$l = h \times H(\epsilon_A || S_A) \times E_B$$

$$g_B = H(\epsilon_B || S_B || E_B)$$
\[ d_B = Z_B \oplus g_B \oplus S_i \]  

(v) Verify if \((C_B = (d_B G - E_B))\) and in the case of success, compute the following parameters:

\[ g_A = H[\text{userid}_A \parallel \text{userid}_B \parallel y(E_A) \parallel y(E_B) \parallel x(l) \parallel S_i] \]  

\[ d_A = H[e_A \parallel c_A \parallel S_i] + c_A \mod q \]  

\[ Z_A = g_A \oplus d_A \oplus S_i. \]  

(vi) Read current time \(T_C^*\), invoke functions TSG and \(E_K\)

\[ T'_a \leftarrow \text{TSG}(T'_c, T_0, \Delta T, \Delta \tau) \]  

\[ m_a \leftarrow E_K(A, B) Z_A \parallel T'_a \]  

(vii) Send the message \(m_a\) to \(S_{UB}\).

(viii) Destroy the session parameters \(e_A, g_A, g_B^*\).

Step 4: As message \(m_a\) is received, \(S_{UB}\) reads arrival time \(T_R^*\).
Invokes the functions \(TSG()\) and \(TSV()\)

\[ T'_b \leftarrow \text{TSG}(T'_B, T_0, \Delta T, \Delta \tau) \]  

\[ S' \leftarrow \text{TSV}(T_R, T_b, T'_B, T_0, \Delta T, \Delta \tau) \]  

Step 5: Verify if \((S' = \text{‘SUCCESS’})\), if yes computes

\[ g'_A = H[\text{userid}_A \parallel \text{userid}_B \parallel x(E_A) \parallel x(E_B) \parallel x(l) \parallel S_i] \]  

\[ d'_A = Z_A \oplus g'_A \oplus S_i \]  

Verify if \((C_A = (d'_A G - E_A))\)  

In this manner, \(S_{UB}\) successfully authenticates \(S_{UA}\) and destroys the ephemeral parameter \(g'_A^*\).  

Step 6: Afterwards, \(S_{UA}\) and \(S_{UB}\) terminate the communication handshake and agree on the unique common secret session key, which is computed as

\[ K_{\text{Session}, A,B} = H[\text{userid}_A \parallel \text{userid}_B \parallel x(E_A) \parallel x(E_B) \parallel x(l) \parallel S_i] \times d_A \parallel d_B \parallel x(l) \parallel S_i \]  

The correctness of the proposed VTE-AKA is derived from the existence of the following equation:

\[ l = h \times H[e_A \parallel c_A \parallel S_i] \times H[e_B \parallel c_B \parallel S_i] \times G \]  

Only legitimate SUs are able to compute the same session key using the same set of aforementioned ECC parameters.

---

**Fig. 2** Registration phase in VTE-AKA scheme
A scalar multiplication of co-factor $h$ with the arithmetic expressions defined over the elliptic curve, proficiently prevents the small subgroup attacks and consequently minimises the occurrence of inappropriate elliptic curve parameters.

### Computational and communicational complexity

The computational complexity of VTE-AKA is segregated into two phases viz. **registration phase** and **AKA phase**. The...
communication complexity is defined as the number of message passes in the AKA phase (Table 4).

Note: The communicational complexity (no. of passes) = 3, number of field inversions = 0, number of field multiplications = 0.

6 Security analysis

The VTE-AKA achieves the desired security attributes of a robust AKA protocol if it resists the known cryptographic attacks viz.

6.1 Key compromise impersonation (KCI)-resilience

In the KCI attack, SU_M masquerades as one of the legitimate SUs (SU_A or SU_B) from an established communication session using the respective long-term private key (c_A or c_B).

Case 1: SU_A’s long-term private key c_A revealed to adversary SU_B and the adversary aspires to impersonate SU_B for establishing a communication session with SU_A.

For impersonating SU_B to SU_A, SU_M needs long-term secret key of SU_B (c_B) for deriving the parameter d_B. Consequently, SU_M needs to compute e_B from E_B and c_B from C_B, where

\[ E_B = H[|e_B| \parallel c_B \parallel S_i] \times G \quad \text{(Ref. (3))} \]

However, as per Conjecture A.1, the ECDLP problem is intractable and H_i is a one-way collision-free hash function. Hence, it is not possible for SU_M to impersonate SU_B by deriving c_B and e_B as both these parameters are random oracle and independently generated at SU_B’s end.

Case 2: SU_A’s ephemeral private key (e_A) is revealed to be an adversary SU_M and the adversary aspires to impersonate SU_A.

Adversary SU_M aspires to impersonate SU_A to other participants using the ephemeral private key (e_A). Nevertheless, SU_M requires a long-term private key (c_A) of SU_A for successful impersonation. Nonetheless, SU_M needs to calculate

\[ c_A = \text{ECDLP}(C_A) \quad \text{(27)} \]

Subsequently, needs to compute

\[ E_A = H_i(e_A \parallel c_A \parallel S_i) \times G \quad \text{(Ref. (1))} \]

However, as per Conjecture A.1, the ECDLP problem is intractable and H_i is a one-way collision-free hash function. Hence, VTE-AKA proficiently resists the KCI attack.

6.2 Known session key security

It is vital for VTE-AKA to satisfy known session key security. So, VTE-AKA incorporates ephemeral private keys (e_A, e_B), ephemeral public keys (E_A, E_B), user session IDs (usid_A, usid_B), long-term private keys (c_A, c_B), session identifiers (d_A, d_B), computed shared secret (e), and unique session identity (S_i) in deriving unique common secret session key (K_{Session_A,B}) and ensures its uniqueness in every established session. It also ensures that at the end of protocol execution, both participants are assured of each other’s identity, and K_{Session_A,B} is based on the data contributed equally by each participant. The uniqueness of session parameters in VTE-AKA ensures that a compromised session key does not reveal other session’s keys.

6.3 Full perfect forward secrecy resilience

Case 1: One of the key pairs (c_A, c_B), (e_A, e_B), (c_A, e_B), and (e_A, c_B) is revealed to adversary SU_M at some instant of time. An ephemeral shared secret for a session is computed as follows:

\[ l = h \times H_i(e_A \parallel c_A \parallel S_i) \times H_i(e_B \parallel c_B \parallel S_i) \times G \quad \text{(Ref. (25))} \]

So, SU_M cannot derive previous sessions’ unique common secret key, as knowledge of random pairs (c_A, c_B) or (e_A, e_B) for the previous session is needed. Still, H_i is a random oracle and the pairs (E_A, C_A, c_A), (H_i[e_A \parallel c_A \parallel S_i], C_A, c_A) are indistinguishable. For SU_M to distinguish (E_A, C_A, c_A), (H_i[e_A \parallel c_A \parallel S_i], C_A, c_A), it must know H_i[e_A \parallel c_A \parallel S_i] and derive c_A = ECDLP(C_A). Based on the same analogy (E_B, C_B, c_B), (H_i[e_B \parallel c_B \parallel S_i], C_B, c_B) are indistinguishable. If SU_M has to distinguish (E_B, C_B, c_B) from (H_i[e_B \parallel c_B \parallel S_i], C_B, c_B), then it must have knowledge of H_i[e_B \parallel c_B \parallel S_i] and should be able to compute c_B ECDLP (C_B). Nonetheless, Conjecture A.1 states that ECDLP is an intractable problem. Though, if SU_M tries to re-construct secret session key from the transmitted messages E_B, m_B, it needs to compute ECDHP (E_A, E_B), which according to Conjecture A.2 is an intractable problem. So, VTE-AKA ensures PFS-R.

6.4 Key replicating-resiliency

Case 1: SU_M strives to establish a fresh communication session using the same keys as that of the test session. VTE-AKA continuously provides message integrity to E_A and m_B and message authentication for Z_i and Z_B in the AKA phase and hence VTE-AKA thwarts the establishment of a fresh session. Thus, the pair of participants SU_M, SU_A exhibit non-matching session parameters including ephemeral private and public keys and unsatisfied decrypted verification equations. The derived session key from a successful fresh session establishment is as follows: (see equation below).

Hence, VTE-AKA proficiently resists key replicating attack.

6.5 Unknown key-share resilience

Case 1: SU_M strives to make participant SU_A believe that it shares a secret with participant SU_A and endeavours to make SU_B believe that it has established a session with SU_A. For impersonating SU_B to SU_A and convincing that it shares a secret with SU_B, SU_M needs to derive a long-term private key (c_B) of SU_M from its long-term public key (C_B). Furthermore, SU_M needs to derive the parameter

\[ H_i(e_B | c_B \parallel S_i) \times ECDHP (E_A, E_B) \] using the message m_B computed during communication handshake between SU_A and SU_B. If SU_M succeeds, to deliver a long-term private key (c_B) of SU_M from its long-term public key (C_B).
in deriving these parameters, then it needs to successfully convince \( \text{SU}_B \) to agree on a pre-selected unique common secret session key (KM). Nevertheless, \( \text{SU}_M \) neither succeeds in deriving \( c_B \) nor in deriving \( H_1[e_B \parallel c_B \parallel S_i] \) because as per Conjecture A.1 deriving \( c_B \) is an intractable problem and deriving \( H_1[e_B \parallel c_B \parallel S_i] \) is not feasible because \( H_1 \) is a one-way collision-free hash function. Besides, computation of unique common secret session key in VTE-AKA involves the unique user session IDs, unique session IDs, long-term private and public keys of \( \text{SU}_A \) and \( \text{SU}_B \), and unique session identifiers (\( d_A, d_B \)), which ensures the non-matching of derived session parameters and hence leads to computation of different session keys at \( \text{SU}_A \)'s and \( \text{SU}_B \)'s ends in the same communication handshake for AKA.

### 6.6 Replay resilience

In the replay attack, \( \text{SU}_M \) masquerades as \( \text{SU}_A \) and strives to establish a fresh session with \( \text{SU}_B \). For stealing the identity of \( \text{SU}_A \), \( \text{SU}_M \) eavesdrops a stream of public messages from the current AKA session between \( \text{SU}_A \) and \( \text{SU}_B \), and replays the same to \( \text{SU}_B \) masquerading as \( \text{SU}_A \) and vice versa. However, the prime order \( n \) of point \( G \) is arbitrarily large and the probability of randomly selecting the same ephemeral private keys \( e_A \) and \( e_B \in \mathbb{Z}_p \) in different sessions is negligible. While \( \text{SU}_M \) masquerading as \( \text{SU}_A \) by replaying the old message \( E_A \) from the previous session, \( \text{SU}_B \) sends a fresh copy of \( m_B \) in the second pass of the fresh session. Consequently, \( \text{SU}_M \) is made to abort as it is unsuccessful in generating parameter \( Z_A \) corresponding to \( E_B \). Similarly, \( \text{SU}_M \) cannot masquerade as \( \text{SU}_B \) since VTE-AKA is a symmetric protocol. Thus, VTE-AKA proficiently resists replay attacks.

### 6.7 Reflection resilience

In a communication handshake, VTE-AKA with three passes is proficient in detecting a legitimate participant by establishing a fresh AKA session with itself. Initiator \( \text{SU} \) is able to verify the identity of a legitimate participant only after the second pass when packet \( m_B \) is verified. Similarly, the identity of the initiator is verified by a legitimate participant only after the third pass when the message packet \( m_A \) is verified. Verification equations for \( C_B \) and \( C_A \) offer signature validation as they contain long-term private keys for both initiator and legitimate participants. Consider the following verification equations:

\[
C_B = (Z_B \oplus g_B \oplus S_i) \times G - E_B = c_B \times G \quad (28)
\]

\[
C_A = (Z_A \oplus g_A \oplus S_i) \times G - E_A = c_A \times G \quad (29)
\]

For a successful AKA execution, these equations hold true, and thus \( \text{SU}_A \) verifies the correctness of message \( m_B \) sent from \( \text{SU}_B \) and \( \text{SU}_B \) verifies the correctness of message \( m_A \) sent from \( \text{SU}_A \) and thus verifying identities of initiators and participants.
Table 1  Notations adopted in VTE-AKA protocol

| Notations | Descriptions |
|-----------|--------------|
| uid_A, uid_B | identity of SU_A and SU_B |
| usid_A, usid_B | unique session identity of SU_A and SU_B |
| S_i | unique session token ID |
| @ | exclusive disjunction |
| || | bitwise concatenation operator |
| C_A, C_B, e_A, e_B | private static and ephemeral keys, respectively |
| C_A, C_B, E_A, E_B | public static and ephemeral keys, respectively |
| m_A, m_B | encrypted message blocks |
| K_{A,B}, K_{T,N,A}, K_{T,N,B} | communication keys |
| E_k[ ] | AES-128 GCM symmetric encryption and decryption functions |
| H[ ], H_1[ ], H_2[ ] | secure one-way hash functions |
| TSG( ) | time stamp generator function |
| TSV( ) | time stamp validator function |
| T_d | symmetric decryption time |
| Ø_ECC | ECC parameters: (q, FR, S, a, b, G, n, h) |
| K_{T,N,A} | reg. phase – symmetric key 1: H[uid_A @ T_A] |
| K_{T,N,B} | reg. phase – symmetric key 2: H[uid_B @ T_B] |
| K_{A,B} | AKA phase – public key 1: C_A @ S_i |
| K_{Session,A,B} | AKA phase – public key 2: C_B @ S_i |
| H[uid_A || usid_A || x(E_A) || x(E_B) || d_A || d_B || x(I) || S_i] | unique common secret session key: |

Table 2  Timing notations adopted in VTE-AKA

| Notations | Descriptions |
|-----------|--------------|
| T_A, T_B | current time stamps registered at SU_A and SU_B |
| T_i | generated time stamp for (Session)_i |
| T_c | current time |
| T_0 | reference time |
| ΔT | unit time interval |
| Δr | relative error between clocks of SU_A and SU_B |
| T_P | elliptic curve polynomial computation time |
| T_PM | elliptic curve point multiplication time |
| T_E | elliptic curve encryption time |
| T_D | elliptic curve decryption time |
| T_s | symmetric encryption time |
| T_d | symmetric decryption time |

Table 3  Abbreviations used in TSG( ) and TSV( )

| Abbreviations | Descriptions |
|---------------|--------------|
| V_c(x,y) | two-dimensional vector-time code |
| T_x* | Session, time-stamp variable |
| G.I.F[ ] | greatest integer function |
| T_0 | reference time |
| ΔT | unit time interval |
| T_R | arrival time |
| Session_i | current session ID |
| flag | Boolean variable |
| event | state of AKA protocol execution |
| status | current state of an entity or procedure |
| observe | state of monitoring or perceiving |
| x(); y() | X and Y coordinates of an elliptic curve point |

6.8 Key control resilience

The secret session key is computed using inputs contributed equally by each participant executing VTE-AKA and hence the protocol ensures that no single participant is able to force the computation of unique session key to a pre-selected value. Refer to the following equation for more clarity:

\[ l = h \times H_i[ea || ca || S_i] \times H_j[eb || cb || S_i] \times G \]  

(Ref. (25))
The implementation results are shown in Table 6. The simulation parameters are shown in Table 5. The implementation results are shown in Table 6. The performance of VTE-AKA is statistically evaluated, analysed, and compared with the schemes proposed by Alhakami et al. [18], Safdar et al. [20] and Jakimoski et al. [21] and VTE-AKA w/o time-stamp in MATLAB R2012a. The chosen performance parameters are malicious node detection probability, PU detection probability, channel utilisation, and average throughput. The simulation results are plotted in Figs. 7–9. For a specific malicious user density, Figs. 7–9 illustrate the impact of a varying number of SUs on the malicious node detection probability. For a given time frame, the mean values of malicious node detection probability for VTE-AKA, its variant without timestamp, Jakimoski et al. [21], Safdar et al. [20], and Alhakami et al. [19] are 0.7535, 0.5539, 0.5654, 0.5519, and 0.5103, respectively. These mean values exhibit a slight increase because the additional legitimate nodes get involved in the decision-making process. Within a given time frame, the VTE-AKA protocol with a standard deviation of 0.1003 is scalable in its MAKA phase and is able to efficiently handle the increased number of communication sessions with augmented $T_{delay}$ and increased Time-to-live (TTL). Furthermore, a scalable timestamp mechanism employed by the VTE-AKA protocol ensures that the asynchronous SUs are competently authenticated and placed as legitimate nodes in the ad hoc CRN network. Consequently, the proposed VTE-AKA protocol outperforms the aforementioned protocols in terms of malicious node detection probability. Statistically VTE-AKA is 32.42% better than its variant, 29.73% better than Jakimoski et al. [21], 32.91% better than Safdar et al. [20], and 43.74% better than Alhakami et al. [19].

For a specific malicious user density, Figs. 10–12 illustrate the impact of varying number of SUs on PU detection probability. Within a given time frame, the mean values of primary node detection probability for VTE-AKA, its variant without timestamp, Jakimoski et al. [21], Safdar et al. [20], and Alhakami et al. [19] are 0.9064, 0.6350, 0.6579, 0.6595, and 0.6425, respectively. The mean values exhibit a small gradual increment as an increase in total SUs. However, the performance metric of the VTE-AKA protocol with a standard deviation of 0.0491 is 42.74% better than its variant without timestamp with a standard deviation of 0.0454, 41.07% better than Alhakami et al. [19], 37.44% better than Safdar et al. [20], and 37.77% better than Jakimoski et al. [21]. This is due to the fact that a competent authentication mechanism in VTE-AKA is employed for asynchronous SUs, and therefore accurate information of sensing data is available for PU detection. Furthermore, a consensus among the participating legitimate SUs is reached within a given time.

Table 4  Computational complexity of VTE-AKA protocol

| Registration | Authentication and key agreement |
|-------------|----------------------------------|
| SU$_n$ | TN | SU$_a$ | SU$_b$ |
| $T_{PM} + T_a + T_H$ | $2T_H + 2T_a + T_H$ | $T_{PM} + T_d + T_H$ | $6T_H + 3T_{PM} + T_E + T_D$ | $6T_H + 3T_{PM} + T_E + T_D$ |

$T_{PM}$ and $T_E$ are incorporated in protocol design. Discrete event simulator OMNET++ with IEEE 802.11b as an extension to support CR networks is used to model the CRAHN environment. In the registration phase, the computational complexity involved in ECC implementation over GF($2^{163}$) and GF($2^{233}$) is $T_{PM} + T_a + T_H$ for a SU and $2T_H + 2T_a + T_H$ for the TN. In the MAKA phase, complexity involved is $6T_H + 3T_{PM} + T_E + T_D$ for both SUs. The computational complexity is defined as the number of passes in a communication session. In the VTE-AKA protocol, it is three as mentioned in Table 4. The simulation time is set in accordance with a time frame of the control phase to facilitate the complete negotiations among the participating SUs. The number of SUs is varied from 50 to 250 and MU density from 3 to 30% during the simulation as listed in Table 5. The performance of VTE-AKA is statistically evaluated, analysed, and compared with the schemes proposed by Alhakami et al. [18], Safdar et al. [20] and Jakimoski et al. [21] and VTE-AKA w/o time-stamp in MATLAB R2012a.

Table 5  Simulation parameters

| Parameter | Settings |
|-----------|----------|
| simulation time | 15 s/iteration |
| network size | 2500 m x 2500 m |
| data packet length | 2000 bytes |
| CR transmission | 100% persistent |
| distributed resource map | enabled |
| MAC protocol | IEEE 802.11b with extension to support CR networks |
| data channels | 2 |
| PU arrival rate | 0.000625/s |
| sensing interval | 25 ms |
| SU arrival rate | 0.005/s |
| number of SUs | 50, 100, 150, 200, 250 |
| number of MUs | 3, 10, 15, 20, 25, 30% |

Table 6  Implementation results

| Registration phase | Elliptic Curve Session Transient Parameters (ESTP) |
|-------------------|--------------------------------------------------|
| computation | 03.39 ms |
| unique session IDs generation | 12.16 ms |
| concatenated ESP symmetric encryption | 05.23 ms |
| concatenated ESP symmetric decryption | 04.72 ms |
| ECC implementation over GF($2^{163}$) | 163-bit, ms 233-bit, ms |
| elliptic curve polynomial computation | 5.23 |
| static private key generation | 2.89 |
| static public key generation | 3.92 |
| authentication and key agreement | 3.012 |
| time stamp generation | 2 x 1.07 ms |
| time stamp validation | 3.98 ms |
| ECC implementation over GF($2^{233}$) | 163-bit, ms 233-bit, ms |
| ephemeral private key generation | 3.012 |
| ephemeral public key generation | 3.169 |
| ephemeral parameters computation | 37.35 |
| concatenated ephemeral parameters | 102.78 |
| asymmetric encryption concatenated ephemeral parameters | 47.31 |
| asymmetric decryption | 83.196 |
| transient authentication | 2.17 ms |

7 Simulations and performance evaluation

7.1 Simulation parameters

The simulation parameters are shown in Table 5.

7.2 Implementation results

The implementation results are shown in Table 6.

7.3 Performance analysis

The VTE-AKA protocol is designed using the ECC arithmetic comprising binary curve curves GF(2) as it offers substantial advantages in performance over prime curves GF(p). Koblitz curves over GF(2)}

$$K_{Session,A,B} = H(\text{usid}_A \parallel \text{usid}_B \parallel x(E_A) \parallel x(E_B)) \parallel d_A \parallel d_B \parallel x(I) \parallel S_i) \quad \text{(Ref. (24))}$$

IET Commun., 2020, Vol. 14 Iss. 20, pp. 3529-3540
© The Institution of Engineering and Technology 2020
frame as the timestamp mechanism in VTE-AKA ensures the well-timed conclusion of authentication and key agreement phase.

For a specific MU density, Figs. 13–15 depict the influence of increasing SUs on channel utilisation. In a time-frame, the mean channel utilisation for the VTE-AKA protocol, its variant without timestamp, Jakimoski et al. [21], Safdar et al. [20], and Alhakami et al. [19] are 0.6037, 0.5755, 0.5386, 0.5157, and 0.4897, respectively. The mean channel utilisation exhibits firstly a steep and then a gradual increment as initially a substantial number of legitimate SUs ensure an efficient authentication process and a number of SUs cross a threshold, number of authentication sessions increase abruptly resulting in frequent session timeouts. Nevertheless, the performance metrics of the proposed VTE-AKA protocol with a standard deviation of 0.3017, in terms of channel utilisation is 23.27% better than Alhakami et al. [19], 4.90% better than its variant without time-stamp with a standard deviation of 0.2903, 17.064% better than Safdar et al. [20] and 12.086% better than Jakimoski et al. [21] and hence outperforms the aforementioned protocols. In VTE-AKA, a substantial size buffer with a greater TTL is involved in the maintenance of the queue at participant nodes. Thus, message delivery probability increases and average latency decreases. Nevertheless, the average waiting time in the buffer and the average latency is decreased due to the introduction of the timestamp mechanism in VTE-AKA, and consequently, channel utilisation increases significantly.

For a fixed malicious user density, Fig. 16–18 depict the influence of varying number of SUs on the CRAHN average throughput. Within a given time frame, the mean values of CRAHN’s throughput for the VTE-AKA protocol, its variant without timestamp, Jakimoski et al. [21], Safdar et al. [20], and Alhakami et al. [19] are 0.5123, 0.4772, 0.4907, 0.4055, and 0.3796, respectively. The throughput means firstly exhibit a steep increase and then a gradual decrease in behaviour as it is inversely proportional to generated messages and associated overheads. As the number of SUs crosses a certain threshold, the collision of
As the total number of SU increases, more number of mutual authentications are required. Hence, the time involved in the authentication phase increases gradually resulting in delayed distributed cooperative sensing. To counter this constraint, VTE-AKA increases the average waiting time in the buffer.

8 Conclusion

In a hostile and dynamic environment, malicious nodes' MUs or selfish SUs notably degrade the performance of CRAHN in terms of network average throughput. Similarly, the MUs and selfish SUs drastically impact the legitimate SU's channel utilisation, MUs' detection probability, and the PUs' detection probability. The adeptness of mutual authentication in CCC transactions is based on the strength of cryptographic primitives and logical time synchronisation among the distributed SUs. To achieve higher security levels while ensuring perfect forward secrecy and avoiding replay attack, a novel vector time encoding technique is proposed and incorporated with a computationally enhanced authenticated key agreement scheme named VTE-AKA. The VTE-AKA encompasses only two passes in a communication handshake and three passes in an established communication session between two intended participants. Consequently, VTE-AKA significantly lowers the communicational overhead. Comprehensive and exhaustive simulations are implemented to study the impact of erratic frequency of unlicensed SUs' and increasing MUs' density on state-of-the-art authenticated key agreement schemes and VTE-AKA. Statistical analysis of simulation results demonstrates the proficiency of VTE-AKA in a hostile environment and proves its robust security against KCI attack, KSK attack and its resilience against key replicating, unknown key share, subsequently providing R-R, KC-R and weak perfect forward secrecy.

9 References

[1] Shi, W., Gong, P.: ‘A new user authentication protocol for wireless sensor networks using elliptic curves cryptography’, Int. J. Distrib. Sensor Netw., 2013, 9, pp. 51–59. Available at https://doi.org/10.1155/2013/730831
[2] Qiu, S., Xu, G., Ahmad, H., et al.: ‘An improved lightweight two-factor authentication and key agreement protocol with dynamic identity based on elliptic curve cryptography’, KSII Trans. Int. Inf. Syst., 2019, 13, (2), pp. 978–1002. Available at https://doi.org/10.3837/tiis.2019.02.027
[3] Roy, S., Chatterjee, S., Das, A.K., et al.: ‘On the design of provably secure lightweight remote user authentication scheme for mobile cloud computing services’, IEEE Access, 2017, 5, pp. 25808–25825. Available at https://doi.org/10.1109/ACCESS.2017.2764913
[4] Yang, Z., Lai, J., Sun, Y., et al.: ‘A novel authenticated key agreement protocol with dynamic credential for WSNs’, ACM Trans. Sen. Netw., 2019, 15, (2), p. 27, Article 22 (February 2019). Available at https://doi.org/10.1145/3303704

[5] Wang, W., Chen, L., Shin, K.G., et al.: ‘Thwarting intelligent malicious behaviors in cooperative spectrum sensing’, Mobile Comput., IEEE Trans., 2015, 14, (11), pp. 2392-2405. Available at https://doi.org/10.1109/TMC.2015.2398446

[6] Srinu, S., Sabat, S.L.: ‘Optimal multinode sensing in a malicious cognitive radio network’, Syst. J., IEEE, 2015, 9, (3), pp. 855–864. Available at https://doi.org/10.1109/JSYST.2013.2283963

[7] Wang, X., Ji, Y., Zhou, H., et al.: ‘Auction-based frameworks for secure communications in static and dynamic cognitive radio networks’, IEEE Trans. Veh. Technol., 2017, 66, (3), pp. 2658-2673. Available at https://doi.org/10.1109/TVT.2016.2581179

[8] Abedi, M.R., Mokari, N., Javan, M.R., et al.: ‘Secure communication in OFDMA-based cognitive radio networks: an incentivized secondary network coexistence approach’, IEEE Trans. Veh. Technol., 2017, 66, (2), pp. 1171-1185. Available at https://doi.org/10.1109/TVT.2016.2555946

[9] Xu, X., He, B., Yang, W., et al.: ‘Secure transmission design for cognitive radio networks with poisson distributed eavesdroppers’, Inf. Forensics Sec., IEEE Trans., 2016, 11, (2), pp. 373–387. Available at https://doi.org/10.1109/TIFS.2015.2500178

[10] Zhang, L., Ding, G., Wu, Q., et al.: ‘Defending against byzantine attack in cooperative spectrum sensing: defense reference and performance analysis’, IEEE Access, 2016, 4, pp. 4011-4024. Available at https://doi.org/10.1109/ACCESS.2016.2593952

[11] Fadlullah, Z.M., Nishiyama, H., Kato, N., et al.: ‘Towards secure spectrum decision’. IEEE J. Sel. Areas Commun., 2013, (11), pp. 1645-1648. Available at https://doi.org/10.1109/JSAC.2013.131119

[12] Liu, S.Q., Hu, B.J.: ‘Analysis of sensing efficiency for cooperative spectrum sensing with malicious users in cognitive radio networks’, IEEE Netw., 2013, 27, (3), pp. 51–56. Available at https://doi.org/10.1109/MNET.2013.6523809

[13] LaMacchia, B., Lauter, K., Miyagin, A.: ‘Stronger security of authenticated key exchange’, ProcSec 2007, Lecture Notes in Computer Science, vol 4784. Springer, Berlin, Heidelberg, 2007. Available at https://doi.org/10.1007/978-3-540-75670-5_1

[14] Leon, H., McLaughlin, S.W., Kim, I.-M., et al.: ‘Secure communications with untrusted secondary nodes in cognitive radio networks’, Wirel. Commun., IEEE Trans., 2014, 13, (4), pp. 1790–1805. Available at https://doi.org/10.1109/TWC.2013.021214.130089

[15] Jo, M., Han, L., Kim, D., et al.: ‘Selfish attacks and detection in cognitive radio ad-hoc networks’, IEEE Netw., 2013, 27, (3), pp. 46–50. Available at https://doi.org/10.1109/MNET.2013.6523808

[16] Qin, Z., Li, Q., Hsieh, G.: ‘Defending against cooperative attacks in cooperative spectrum sensing’, Wirel. Commun., IEEE Trans., 2013, 12, (6), pp. 2680, 2687. Available at https://doi.org/10.1109/TWC.2013.041913.120516

[17] He, X., Dai, H., Ning, P.: ‘A byzantine attack defender in cognitive radio networks: the conditional frequency check’, IEEE Trans. Wirel. Commun., 2013, 12, (5), pp. 2512–2523. Available at https://doi.org/10.1109/TWC.2013.031313.121551

[18] Tang, H., Yu, F.R., Huang, M., et al.: ‘Distributed consensus-based security mechanisms in cognitive radio mobile ad-hoc networks’, Commun., IET, 2012, 6, (8), pp. 974, 983. Available at https://doi.org/10.1049/iet-com.2010.0553

[19] Allhakami, W., Mansour, A., Safdar, G.A.: ‘Shared-key based secure MAC protocol for CRNs’. Next Generation Mobile Apps Services and Technologies (NGMAST) 2014 Eighth Int. Conf. on, Oxford, UK, 10–12 September 2014, pp. 266,271. Available at https://doi.org/10.1109/NGMAST.2014.30

[20] Safdar, G.A., Albernani, S., Aslam, M., et al.: ‘Prevention against threats to self-coexistence - A novel authentication protocol for cognitive radio networks’. Wireless and Mobile Networking Conf. (WMNC) 2014 7th IFIP, Vilamoura, Portugal, 20-22 May 2014, pp. 1-6. Available at https://doi.org/10.1109/WMNC.2014.6878857

[21] Jakimoski, G., Subbalakshmi, K.P.: ‘Towards secure spectrum decision’. Communications 2009. ICC09. IEEE Int. Conf. on, Dresden, Germany, 14–18 June 2009, pp. 1, 5. Available at https://doi.org/10.1109/ICC.2009.5199536

[22] Li, X., Peng, J., Obaidat, M.S., et al.: ‘A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems’, IEEE Syst. J., 2020, 14, (4), pp. 39–50. Available at https://doi.org/10.1109/JSYST.2019.2899580

[23] He, X., Dai, H., Ning, P.: ‘HMM-based malicious user detection for robust collaborative spectrum sensing’. IEEE J. Sel. Areas Commun., 2013, 31, (11), pp. 2196–2208. Available at https://doi.org/10.1109/JSAC.2013.131119

[24] Sun, C., Liu, J., Xu, X., et al.: ‘A privacy-preserving mutual authentication resisting doS attacks in VANETs’, IEEE Access, 2017, 5, pp. 24012–24022. Available at https://doi.org/10.1109/ACCESS.2017.2768499

[25] Reddy, A.G., Das, A.K., Yoon, E.J., et al.: ‘A secure anonymous authentication protocol for mobile services on elliptic curve cryptography’, IEEE Access, 2016, 4, pp. 4394–4407. Available at https://doi.org/10.1109/ACCESS.2016.2596292