Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2009, Article ID 243956, 11 pages
doi:10.1155/2009/243956

Research Article

Performance Analysis of Novel Randomly Shifted Certification Authority Authentication Protocol for MANETs

G. A. Safdar and M. P. O'Neill (nee McLoone)

The Institute of Electronics, Communications and Information Technology (ECIT), Queen's University of Belfast, Northern Ireland Science Park, Queen's Road, Queen's Island, Belfast BT3 9DT, UK

Correspondence should be addressed to G. A. Safdar, ghazanfar.safdar@beds.ac.uk

Received 14 August 2008; Revised 19 March 2009; Accepted 3 June 2009

Recommended by Kameswara Namuduri

The provision of security in mobile ad hoc networks is of paramount importance due to their wireless nature. However, when conducting research into security protocols for ad hoc networks it is necessary to consider these in the context of the overall system. For example, communicational delay associated with the underlying MAC layer needs to be taken into account. Nodes in mobile ad hoc networks must strictly obey the rules of the underlying MAC when transmitting security-related messages while still maintaining a certain quality of service. In this paper a novel authentication protocol, RASCAAL, is described and its performance is analysed by investigating both the communicational-related effects of the underlying IEEE 802.11 MAC and the computational-related effects of the cryptographic algorithms employed. To the best of the authors' knowledge, RASCAAL is the first authentication protocol which proposes the concept of dynamically formed short-lived random clusters with no prior knowledge of the cluster head. The performance analysis demonstrates that the communication losses outweigh the computation losses with respect to energy and delay. MAC-related communicational effects account for 99% of the total delay and total energy consumption incurred by the RASCAAL protocol. The results also show that a saving in communicational energy of up to 12.5% can be achieved by changing the status of the wireless nodes during the course of operation.

1. Introduction

Network security has received critical attention from both academia and industry in recent years. As data networks scale and become more pervasive, network intrusion and attacks have become severe threats to network users [1]. Compared to their wired counterpart, wireless networks are prone to security attacks ranging from passive eavesdropping to active interfering. The open access to the radio interface in wireless networks exposes the content of communication over the wireless link between two mobile units, and between mobile units and the wired network. Such openness also gives an intruder the option to masquerade as a legitimate user. The key security requirements of confidentiality, integrity, authentication and nonrepudiation need to be provided to offer proper protection to wireless links. In principle these features can be achieved through robust key management and cryptographic techniques. Authentication which can be realized by either public (asymmetric) or private (symmetric) key cryptography is of particular importance as it provides a first line of defense against attacks and forms the basis for achieving the other security goals of integrity and confidentiality. An authentication protocol involves a sequence of message exchanges which verify the identities of nodes in a distributed system wishing to communicate [2, 3]. A trusted third party (TTP) that is mutually trusted may or may not be involved as part of the authentication protocol. Public key cryptography (PKC) has been widely accepted as an effective mechanism for providing the fundamental security requirements [4]. It involves a TTP which holds public key certificates acting as a certification authority (CA).

Much research has been conducted into authentication techniques for ad hoc networks that distribute the CA functionality to a set of nodes in the network in a process known as threshold cryptography [5–7]. This method has several disadvantages such as the compromise of the entire network if the collector node is compromised (collector nodes collect the partial certificates generated from different server nodes before generating the complete certificate to be sent), the lack of network growing rules and the adverse effect on the ad hoc network life span due to partial certificate collection and complete certificate generation times. Distributed CA schemes are also believed to be too computationally expensive. Identity- (ID-) based cryptography [8], which eliminates the need for public key certificates and uses participating node IDs as the public keys, can also be used to achieve security in ad hoc networks. This method is more bandwidth efficient than PKC, which requires additional messages for the distribution and exchange of public keys before any cryptographic action can take place [9, 10]. In ID-based systems, a recipient needs a personal secret key generated by a Private Key Generator (PKG) against the recipient's ID to decipher encrypted text. The recipient sends its ID encrypted with the master public key to the PKG. The PKG then generates the personal secret key and sends it to the recipient encrypted with the same master public key. Since this master public key is generated and sent by the PKG to all nodes during the setup phase of an ID-based system, the personal secret key for a particular recipient can also be retrieved by any other node possessing the master public key. Nodes themselves can act as CAs to collect and issue public key certificates on demand to 1 and 2 hop neighbours [11], using broadcast messages to establish a chain of trust across the network. Security in ad hoc networks can also be achieved by clustering of the network with one predefined node in each cluster acting as a cluster head, which executes all administrative functions of the cluster and holds a share of the network secret (key), used for certification, [12]. Soft decision processes have also been used in wireless sensor networks to achieve security by intrusion detection [13]. Unlike authentication, however this is a second line of defense and is carried out by observing several attacks such as collisions, unfairness and exhaustion.

Many of the security solutions that have been proposed for ad hoc networks, such as [5, 10, 12, 14], fail to consider the underlying Medium Access Control (MAC) characteristics; the nodes in an ad hoc network need to strictly obey the rules of the MAC to transmit the security related messages while still maintaining the necessary quality of service (QoS). In this research a novel randomly shifted certification authority authentication protocol (RASCAAL) [15] has now been developed by taking into account the constraints of ad hoc networks in addition to the radio technology MAC related characteristics. The intrusion detection solution proposed by Ren and Liang [13] does consider the underlying MAC characteristics, however whereas the focus of their work is to observe attacks, the RASCAAL protocol is a first line of defence and aims to prevent attacks through authentication.

RASCAAL is based on the proven concept of public key cryptography and provides node authentication by using ACTIVE CAs, which provide an assurance of a node's public key certificate to any other node in the network. RASCAAL forms dynamic random clusters with no prior knowledge of the cluster head by a random shift in the role of ACTIVE CA to any other IDLE CA in the network at the end of a transaction. This differs from the threshold cryptography approach and the concept of permanent clusters with predefined cluster heads. Knowledge of predefined cluster heads makes the network more vulnerable to attack. The salient feature of RASCAAL's security is the very short life span of a randomly formed dynamic cluster, which is equivalent to the duration of the current transaction. This short existence and randomness in cluster formation with no prior knowledge of the cluster head makes the system robust and difficult to attack.

The research described in this paper builds on previous work by the authors in which the security of RASCAAL is thoroughly analysed using BAN logic [15]. Here the performance of RASCAAL is analysed taking into account the effects of the underlying MAC. This is necessary in order to illustrate the effects of security protocols in the context of the overall wireless ad hoc network. The protocol is implemented on top of an IEEE 802.11b Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) scheme [16] and simulated using the OPNET Modeller simulation tool [17]. RASCAAL's communication and computational delay in addition to its energy consumption are investigated. The novel RASCAAL protocol [15] is described in detail in Section 2. Section 3 outlines the simulation setup while a performance analysis is provided in Section 4. Finally conclusions are discussed in Section 5.

2. Randomly Shifted Certification Authority Authentication Protocol (RASCAAL)

RASCAAL has been specifically designed for heterogeneous ad hoc networks in which some of the nodes have additional hardware/software resources over other nodes. The resource enriched nodes with higher buffer and computational resources can act as CA nodes and the density of non-CA (resource constrained) nodes in the network will determine the number of CA nodes required to provide security services. This results in a linear relationship between network security and a threshold number of CA nodes (minimum value $\geqq 2$). A larger number of available CA nodes results in increased randomness thereby increasing the security of the network. To start the network activity, any CA node sensing that the channel is idle for a length of time equal to the Point Coordination Function interframe Space (PIFS) can become an ACTIVE CA by transmission of an ACTIVE_CA_MESSAGE. Subsequently, the current ACTIVE CA randomly selects the ID of a future ACTIVE CA before this role is shifted by the transmission of a TRANSFER_CA_OWNERSHIP message. The underlying MAC provides prioritized access to the medium to transmit an ACTIVE_CA_MESSAGE management (or control) frame. To take control of the medium an ACTIVE CA waits for PIFS, as compared to other non-CA nodes which wait for a Distributed Coordination Function interframe Space (DIFS), where PIFS < DIFS. The ACTIVE CA node engages in servicing public key certificate requests from other IDLE CAs (a CA node which has not become an ACTIVE CA yet) and non-CA nodes in the network. In the following sections, RASCAAL messages have been designed to obey the underlying IEEE 802.11b frames and MAC requirements. Table 1 contains all the notations used in the subsequent description of the RASCAAL protocol. The assumptions made to facilitate the analysis of RASCAAL are given as follows.

(i) The protocol has been developed for heterogeneous networks in which some nodes are resource enriched in comparison to other nodes.

(ii) The IDs of all (CA and non-CA) nodes are known prior to deployment, for example, a small military network.

(iii) Ideal wireless channel conditions are assumed in order to avoid any packet loss and retransmissions (non ideal channel conditions have no direct effect on the performance analysis of security protocols; however, they would result in MAC enabled retransmissions due to packet losses, thereby further increasing the communicational losses).

(iv) The protocol assumes that at least one non-CA node is always in range of an ACTIVE CA node so that it can initiate communication with another non-CA node.

(v) CA and non-CA nodes have synchronized timers (non-CA nodes can extract time values from the broadcast messages of the CA nodes to achieve synchronization for the duration of the randomly formed cluster).

2.1. RASCAAL (Initialization). As part of initialization, which provides key management for the protocol, offline storage of all participating node public keys can be performed. The ACTIVE CA node can transfer the image of stored public keys to other IDLE CA nodes upon request. Alternatively, the number of public keys to be stored can be divided, and keys up to a certain number (depending on the density of non-CA nodes), can be stored per CA. On demand transfer of public keys can take place between both the ACTIVE CA and other IDLE CA nodes employing multi-hop operation in the network. Additionally, dynamic key management can take place where nodes can listen for an ACTIVE_CA_MESSAGE, as shown in (1), and upload their public keys in a SEND_PUBLIC_KEY frame, outlined in (2). Thanks to initialization, RASCAAL does not require synchronisation of the public key certificates maintained by the CA nodes. Hashing is employed in the RASCAAL protocol to provide message integrity. Keyed hashing could also be utilised to provide authentication of nodes in addition to message integrity and the associated key could be stored in both CA and non-CA nodes at the time of initialization.

Table 1: RASCAAL notations.

Notation	Usage
CA_i	CA node
N_j	Non-CA node
IDCA_i	CA node's ID
$\text{ID}N_j$	Non-CA node's ID
T_s	Time stamp value
H	Hash value
$\text{PUB}_{\text{CA}i}$, $\text{PRI}_{\text{CA}i}$	CA node's public and private key
PUB_{Nj}, PRI_{Nj}	Non-CA node's public and private key
E-PUB	Encrypted with public key (CA/non-CA nodes)
E-PRI	Encrypted with private key (CA/non-CA nodes)
BCAST_COUNT	Broadcast count value
X	Securely communicated message

Table 2: RASCAAL simulation parameters.

Parameter	Value
Slot_Time	20 μs
SIFS	10 μs
PIFS	Slot_Time + SIFS
DIFS	2 ∗ Slot_Time + SIFS
Data rate	11 Mbps
Layer 2 payload	8000 bits
Simulated time	600 seconds

Message 1: ACTIVE_CA_MESSAGE:

$$\begin{gathered}\text{CA}_i \longrightarrow N_{(j\cdots n)}, \text{CA}_{(i\cdots n)} : \\ [\text{IDCA}_i, \text{IDCA}_{i-1}, \text{PUB}_{\text{CA}i}, T_s, \text{BCAST\_COUNT}, H].\end{gathered} \tag{1}$$

Message 2: SEND_PUBLIC_KEY:

$$N_j \longrightarrow \text{CA}_i : \left[\text{ID}N_j, \text{PUB}_{Nj}, T_s, H\right]\text{E-PUB}_{\text{CA}i}. \tag{2}$$

In (1) and (2), T_s is the time stamp value and H is the hash of the message for integrity checking. PUB specifies the public key (CA or non-CA node). In (1), IDCA_i is the ID of the current ACTIVE CA to which the ownership has been transferred from CA_{i-1}, thus CA_i forms the new cluster for the current transaction. IDLE CA and non-CA nodes can identify the current ACTIVE CA from the ACTIVE_CA_MESSAGE. BCAST_COUNT is incremented each time the message is rebroadcast by other intermediate IDLE CA nodes up to a maximum value (usually equal to the number of CA nodes in the network) to limit the number of rebroadcasts; all intermediate IDLE CA nodes must concatenate their IDs in the message before rebroadcasting.

2.2. RASCAAL (Public Key Request/Reply and Secure Transaction). At the end of initialization any non-CA node who wishes to communicate with another node requests the

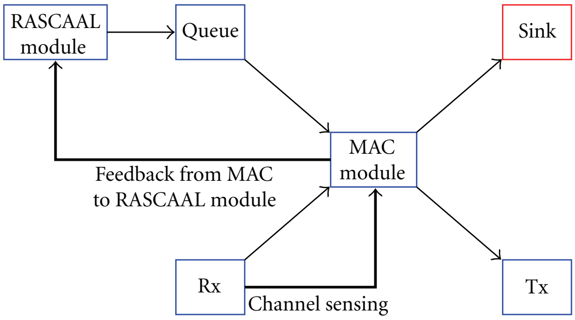


Figure 1: RASCAAL node diagram.

Table 3: Performance of encryption algorithm and hash function designs.

Algorithm/hash functions	Time (μs)	Clock speed (KHz)	Area (gates)	Clock cycles	Power (μW)	Total energy (nJ)
Encryption algorithms:						
Rabin: 512 bits, 0.18 μm CMOS	778	500	21106	389	155	120.6
NtruEncrypt [19]: 167 bits, 0.13 μm CMOS	866	500	16200	—	118.7	102.79
ECC [20]: 134 bits, 0.13 μm CMOS	2.1E05	200	6103	42768	13	2780
HECC [20]: 134 bits, 0.13 μm CMOS	5.4E05	200	7652	109296	17	9290
RSA [22]: 1024 bits, 0.18 μm CMOS	1750	460	—	803822	8.3E05	1.45E06
Hash functions:						
SHA-256 [21]: 256 bits, 0.35 μm CMOS	11280	100	10868	1128	8.79	99
SHA-1 [23]: 128 bits, 0.13 μm CMOS	0.00572	500	4276	405	26.73	21.65
MD5 [21]: 128 bits, 0.35 μm CMOS	7120	100	8001	712	4.74	33.75

destination node's public key from the current ACTIVE CA in a PUBLIC_KEY_REQUEST frame, as outlined in (3). The protocol assumes that at least one of the non-CA nodes is in the range of a current ACTIVE CA node so that it can initiate a secure communication with another non-CA node. The ACTIVE CA will either have the required public key certificate itself or can request it from other IDLE CA nodes and will complete the transaction by transmission of a PUBLIC KEY REPLY, as shown in (4).

Message 3: PUBLIC_KEY_REQUEST:

$$N_j \longrightarrow \mathrm{CA}_i :$$

$$\left[\mathrm{ID}N_j, \mathrm{ID}N_{j+1}, T_s, \left(\mathrm{ID}N_j, \mathrm{ID}N_{j+1}, T_s\right)\text{E-PRI}_{Nj}\right]\text{E-PUB}_{\mathrm{CA}i}. \tag{3}$$

Message 4: PUBLIC_KEY_REPLY:

$$\mathrm{CA}_i \longrightarrow N_j : \left[\mathrm{PUB}_{Nj+1}, T_s, \mathrm{IDCA}_i, H\right]\text{E-PUB}_{Nj}. \tag{4}$$

N_j is the node requesting N_{j+1}'s public key and PRI_{Nj} is node N_j's private key. N_j can successfully initiate a secure transaction with N_{j+1} using its public key. A message "X" can be sent in a SECURE_TRANSACTION_MESSAGE, as described in (5) by node N_j to node N_{j+1} in which N_j also supplies its public key for two way communication.

Message 5: SECURE_TRANSACTION_MESSAGE:

$$N_j \longrightarrow N_{j+1} : \left[\mathrm{PUB}N_j, T_s, X\right]\text{E-PUB}\, N_{j+1}. \tag{5}$$

2.3. RASCAAL (CA Ownership Transfer). At the end of a successful transaction, the current ACTIVE CA randomly selects the ID of any other available IDLE CA and shifts the CA ownership by a TRANSFER_CA_OWNERSHIP message, shown in (6). If there is inactivity in the channel with no communication between the nodes and any current ACTIVE CA for a time period of TRANSFER_CA_OWNERSHIP frame + 2 ∗ max IEEE 802.11 MAC frame, the ACTIVE CA ownership will still be shifted for increased security. The new ACTIVE CA to which the ownership has

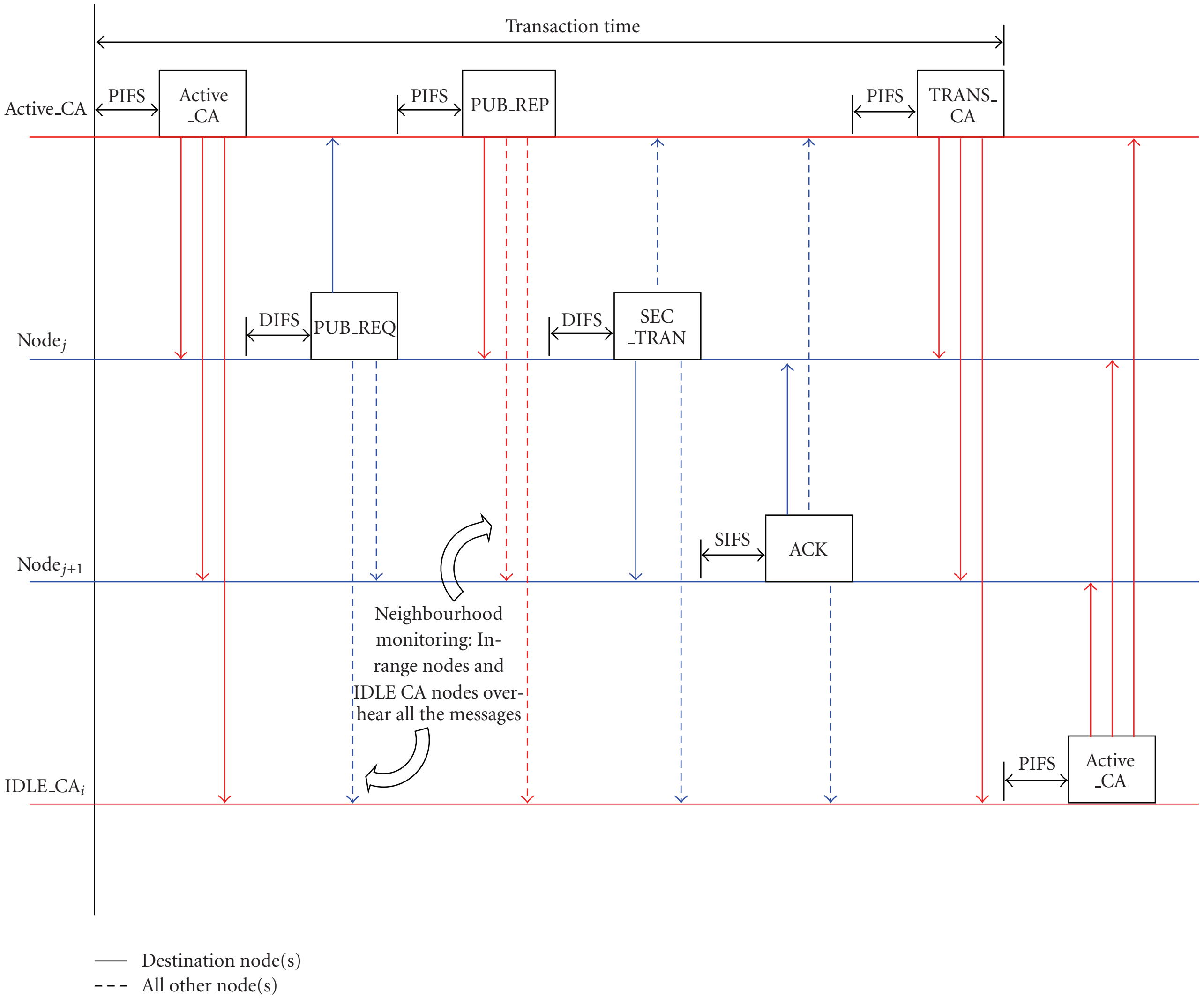


Figure 2: Complete transaction time (diagram does not include back off performed by non-CA nodes).

just been transferred announces its CA ownership by an ACTIVE_CA_MESSAGE (1). This results in the formation of a temporary cluster with a randomly selected cluster head for a duration equal to the current transaction. The broadcast nature of the message, and additionally the presence of both the old ACTIVE CA ID and newly elected ACTIVE CA ID helps to identify any malicious ACTIVE CAs.

Message 6: TRANSFER_CA_OWNERSHIP:

$$\begin{aligned}&\mathrm{CA}_{i-1} \longrightarrow \mathrm{CA}_i :\\ &[\mathrm{IDCA}_{i-1}, \mathrm{IDCA}_i, T_S,\\ &\quad (\mathrm{IDCA}_{i-1}, \mathrm{IDCA}_i, T_s)\text{E-PUB}_{\mathrm{CA}i}, \text{BCAST\_COUNT}, H].\end{aligned} \tag{6}$$

The BCAST_COUNT value is used to limit the number of rebroadcasts, thus lowering the communicational related energy consumption of a node. This value is only found in the messages sent by the CA nodes (ACTIVE and IDLE CA nodes). The CA ownership transfer message is rebroadcast by the intermediate IDLE CA nodes with an increment in the BCAST_COUNT value. The BCAST_COUNT value is reset by the destination node or once it reaches a maximum value which is equal to the number of available CA nodes in the network.

2.4. RASCAAL (Node/CA ID Revocation). Both ACTIVE and IDLE CA nodes have an information base of already associated nodes and nodes that may potentially join the network. As such, only valid CA nodes know the other available CA nodes and the corresponding maximum BCAST_COUNT value, and therefore, any rogue CA node or malicious activity can be detected if the BCAST_COUNT value has gone beyond the maximum value. IDLE CA nodes always concatenate their own IDs before doing any rebroadcasting for increased security and neighbourhood monitoring, which helps to identify compromised or malicious CA nodes. If any fake or duplicate non-CA node ID is found, the ACTIVE

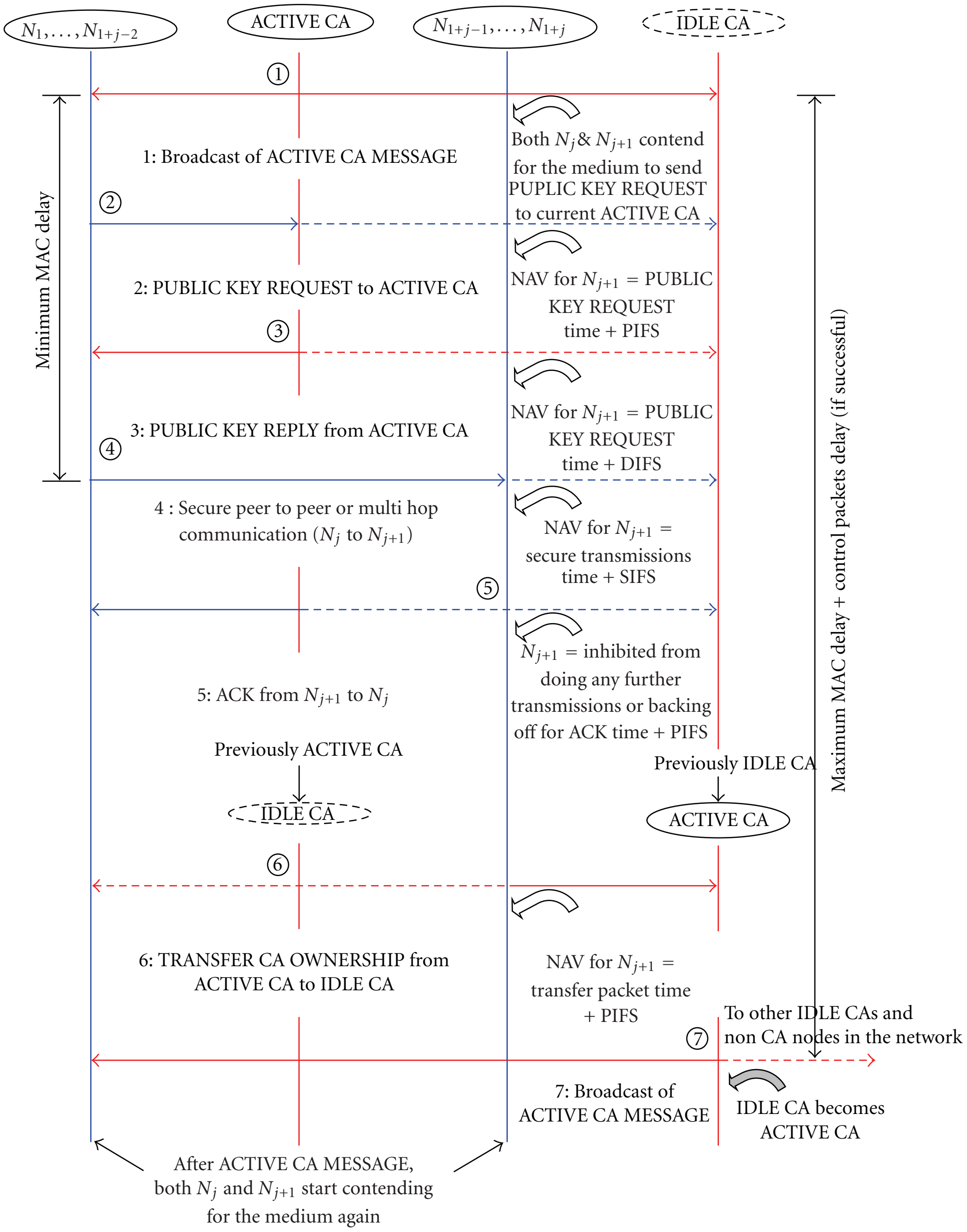

N1, . . . , N1+j−2
ACTIVE CA
N1+j−1, . . . , N1+j
IDLE CA
1
1: Broadcast of ACTIVE CA MESSAGE
Both Nj & Nj+1 contend for the medium to send PUPLIC KEY REQUEST to current ACTIVE CA
2
2: PUBLIC KEY REQUEST to ACTIVE CA
NAV for Nj+1 = PUBLIC KEY REQUEST time + PIFS
3
3: PUBLIC KEY REPLY from ACTIVE CA
NAV for Nj+1 = PUBLIC KEY REQUEST time + DIFS
4
Minimum MAC delay
4 : Secure peer to peer or multi hop communication (Nj to Nj+1)
NAV for Nj+1 = secure transmissions time + SIFS
5
Nj+1 = inhibited from doing any further transmissions or backing off for ACK time + PIFS
5: ACK from Nj+1 to Nj
Previously ACTIVE CA
IDLE CA
Previously IDLE CA
ACTIVE CA
6
6: TRANSFER CA OWNERSHIP from ACTIVE CA to IDLE CA
NAV for Nj+1 = transfer packet time + PIFS
To other IDLE CAs and non CA nodes in the network
7
7: Broadcast of ACTIVE CA MESSAGE
IDLE CA becomes ACTIVE CA
Maximum MAC delay + control packets delay (if successful)
After ACTIVE CA MESSAGE, both Nj and Nj+1 start contending for the medium again

Figure 3: RASCAAL message sequence chart.

CA node can access the medium with priority to revoke that particular node ID in a NODE_ID_REVOKE message, described in (7). Similarly any old ACTIVE CA who has just shifted the CA ownership or other IDLE CA nodes can detect and announce a fake CA ID using a CA_ID_REVOKE message, as shown in (8).

Message 7: NODE_ID_REVOKE:

$$\mathrm{CA}_i \longrightarrow N_{(j\cdots n)}, \mathrm{CA}_{(i\cdots n)} : \left[\mathrm{ID}N_j, \mathrm{IDCA}_i, T_s, \mathrm{BCAST\_COUNT}, H\right]\text{E-PRI}_{\mathrm{CA}i}. \tag{7}$$

Message 8: CA_ID_REVOKE:

$$\mathrm{CA}_{i-1} \longrightarrow N_{(j\cdots n)}, \mathrm{CA}_{(i\cdots n)} : [\mathrm{IDCA}_i, \mathrm{IDCA}_{i-1}, T_s, \mathrm{BCAST\_COUNT}, H]\text{E-PRI}_{\mathrm{CA}i-1}. \tag{8}$$

In (7) and (8), $\mathrm{ID}N_j$ and IDCA_i are the malicious node and CA IDs, respectively. Both messages are encrypted with the private keys of either the ACTIVE CA or the old ACTIVE CA (which has just shifted the ownership); thus only the nodes possessing the relevant public key pairs can decrypt the messages. RASCAAL does not provide provision for the

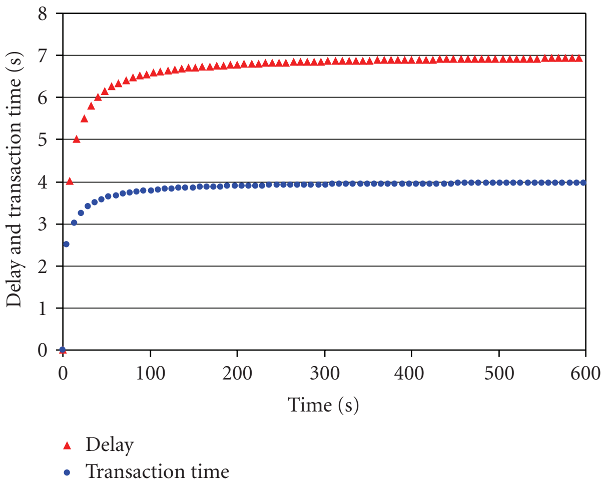


Figure 4: Transaction time versus delay.

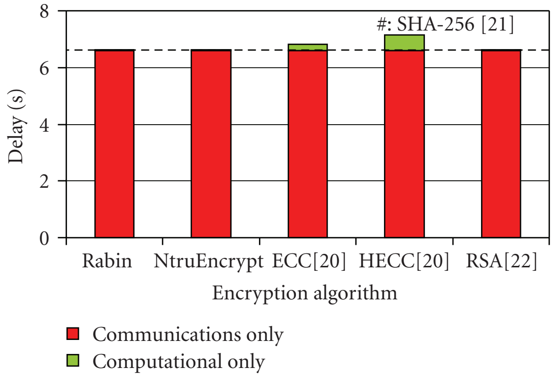


Figure 5: Communicational (average) + computational delay.

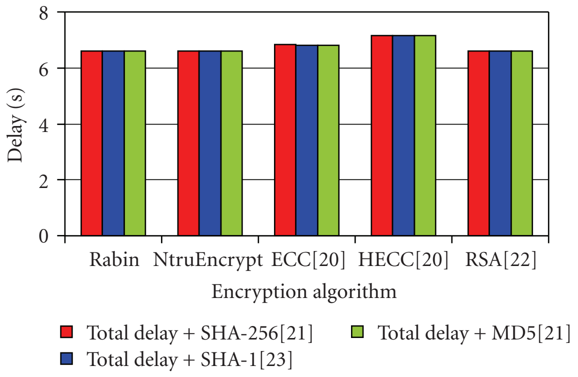


Figure 6: Communicational (average) + computational delay for different hash functions.

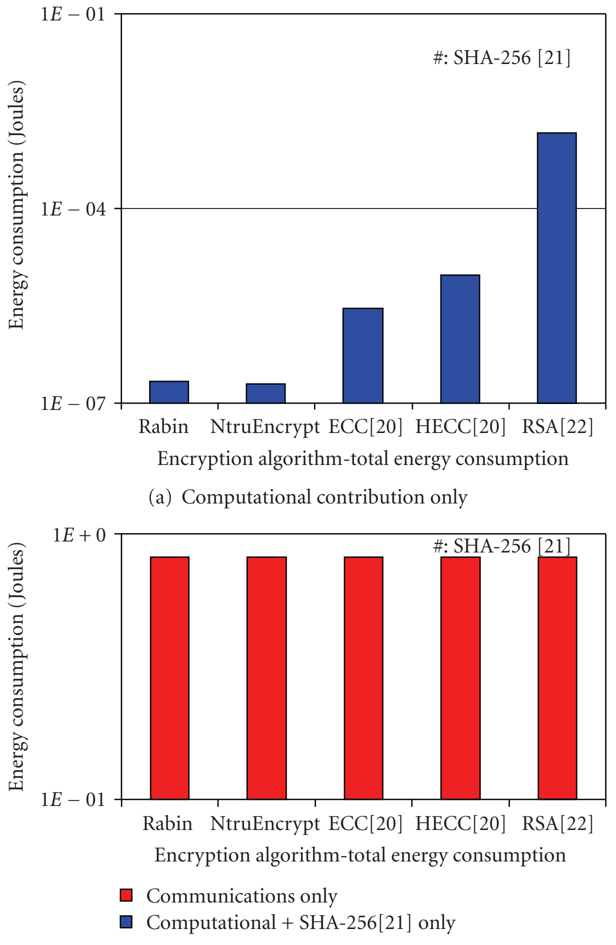
redemption of compromised CA nodes, rather a CA node is declared malicious by revocation (CA_ID_REVOKE).

2.5. RASCAAL Security Analysis. An analysis of the RASCAAL protocol using Burrows-Abadi-Needham (BAN) [18] logic was conducted. Security or cryptographic protocols can have flaws that enable attackers to influence the protocol without requiring the appropriate key, or where the cryptographic algorithms used by the protocol can be broken. This motivates the need for a formal validation of cryptographic protocols because informal methods are not adequately able to analyse security flaws. BAN is a formal logic that can be used to formally describe authentication protocols, and protocols can be verified by following BAN logic definitions and postulates. BAN logic defines a series of predicates, together with mapping instructions for converting message exchanges into formulas, thereby enabling analysis of the knowledge and beliefs that peer entities obtain during an authentication dialogue. A detailed description of the BAN logic analysis of RASCAAL is presented in previous research by the authors [15]. This analysis illustrated that RASCAAL can successfully authenticate nodes and achieves secure communication between them by assuring the ownership of public key certificates. Additionally, the protocol can successfully identify and revoke any malicious or rogue nodes. Confidentiality in RASCAAL is obtained by encryption using public and private keys and hash values have been employed where needed to provide integrity. Finally, time stamp values provide nonrepudiation since all the node timers and timing values are synchronised with the CA clock.

Even if a malicious CA node copies the ID and steals the public key of a valid CA node from the *ACTIVE_CA_MESSAGE*, it can not behave normally because a malicious CA node will not be able to decrypt the *SEND_PUBLIC_KEY, PUBLIC_KEY_REQUEST* or *TRANSFER_CA_OWNERSHIP* messages. These messages can only be decrypted by a valid CA's corresponding private key which is only maintained by the valid CA node (loaded at the initialization stage). Additionally neighbourhood monitoring can help to distinguish and isolate malicious CA nodes. All CA nodes concatenate their own IDs when rebroadcasting and a malicious CA node which has copied the ID of a valid CA node will not know what other CA nodes are available in the network and to whom the CA role can be shifted in the *TRANSFER_CA_OWNERSHIP* message.

3. Simulation Details

Using the OPNET Modeller discrete event simulator tool [17], RASCAAL was implemented on top of a CSMA/CA scheme and simulations were performed for an IEEE 802.11b network consisting of two CA nodes and two non-CA nodes. This number of CA and non-CA nodes is sufficient to implement and analyse all the features of RASCAAL. The main simulation parameters are given in Table 2. Both CA and non-CA nodes have independent RASCAAL modules and a queue to imitate the behaviour of layer 3 and above, and to generate all the security related messages. Depending upon the type of packets received by the layer 2 MAC module, the RASCAAL module only generates packets when invoked by the MAC. The complete node diagram is shown in Figure 1. The MAC process implemented in the MAC module of Figure 1 differs for both CA and non-CA nodes. Compared to non-CA nodes, CA nodes do not perform any back off and always have prioritized medium



(a) Computational contribution only

(b) Encryption algorithm—total energy consumption

Figure 7: Communicational (average) + computational energy consumption.

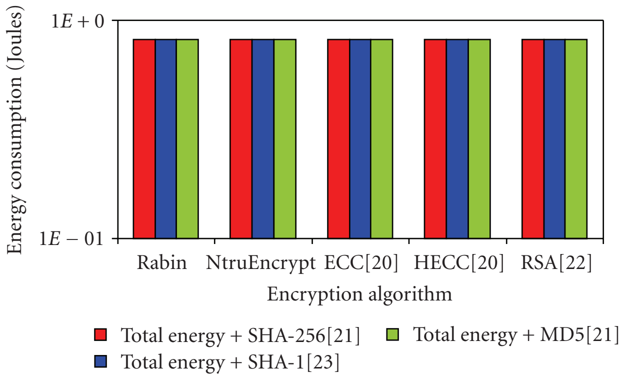


Figure 8: Communicational (average) + computational energy consumption for different hash functions.

access. The CSMA/CA MAC model employed with the RASCAAL protocol utilises a uniform distribution for the number of back off times and an exponential distribution for the contention window. The back off timer is uniformly distributed because the back off value is computed from a uniform distribution. Additionally, a simulation time of 600 seconds is chosen because the uniform distribution has settled to a steady state after this length of time.

Since the focus of this research is to analyse the effects of the underlying MAC on the performance of RASCAAL and vice versa, simulations were performed assuming ideal channel conditions. Additionally, our simulations of the RASCAAL protocol assume that at least one of the non-CA nodes is in the range of a current ACTIVE CA node to initiate a secure communication with another non-CA node. Since wireless ad hoc networks are typically resource constrained, an investigation was carried out into the communicational and computational delay and energy consumption overheads incurred by the RASCAAL protocol. The underlying medium access control protocol was solely responsible for all the communicational losses (delay and energy consumption). The computational overhead was investigated by studying encryption and hash function hardware architectures that were previously proposed in literature for use in resource constrained applications [19–23] and which could be utilised in conjunction with the RASCAAL protocol. The encryption and hash function designs utilized in the analysis are summarized in Table 3. The performance results provided for Rabin's scheme are based on current work being conducted by the authors. Also, it should be noted that the results presented for the NTRUEncrypt algorithm are for a design that offers a security level of 76 bits. In February 2009, this algorithm was approved for standardization by the IEEE at security levels ranging from 112 bits to 256 bits [24]. To date, however, published research into the performance analysis of NTRUEncrypt has focused on a 76-bit security level.

The communicational energy consumption was calculated by assuming that a station's operating current was 290 mA for transmitting (Tx), 205 mA for receiving (Rx) and 62 mA for doze or sleep mode (Prism 2.5, 3.3 volts, IEEE 802.11b network interface card). It was then calculated as

$$P = V * I_{\text{Total}}, \tag{9}$$

where the total current value, I_{Total}, is computed for one complete run of simulation and is given by $I_{\text{Total}} = I_{\text{Sleep}} + I_{\text{Rx}} + I_{\text{Tx}}$. Values for I_{Sleep}, I_{Rx}, and I_{Tx} are obtained by multiplying the total sleep, Rx-On and Tx-On time with the respective current values given for the specified Prism network interface card. The "On" time value is computed by dividing the length of a layer 2 packet by the datarate. The overall energy consumption value was obtained by the summation of communicational and computational values. Similarly, the overall delay value was the addition of communicational and computational delay values. All the delay and energy consumption results are discussed and analysed in the following section.

4. RASCAAL Performance Analysis and Discussion

4.1. Transaction Time versus Node Access Delay. The complete transaction time for RASCAAL as illustrated in Figure 2 and described in (10) is the time spent between the transmission of an ACTIVE_CA_MESSAGE by the current ACTIVE CA and a TRANSFER_CA_MESSAGE, when the role of the ACTIVE CA is randomly shifted to any other IDLE CA available in the network (the ID of an IDLE CA from the list of available IDs is randomly chosen from a uniform distribution). A message sequence chart showing the message exchanges between both non-CA and CA nodes, and ACTIVE and IDLE CA nodes, is shown in Figure 3. This chart gives a detailed explanation of the RASCAAL protocol and illustrates the minimum and maximum MAC delay incurred by successful and contending non-CA nodes, respectively. A contending non-CA node incurs maximum MAC delay due to the very nature of CSMA/CA. It waits by extracting the value of the Network Allocation Vector (NAV) from the packets (messages) not destined to itself. A detailed delay analysis is provided in Section 4.2:

$$\begin{aligned}
&\text{Transaction time}\\
&\quad= [(3 * \text{PIFS}) + \text{SIFS} + (2 * \text{DIFS})\\
&\qquad + (\text{ACTIVE\_CA\_MESSAGE}\\
&\qquad\quad + \text{TRANSFER\_CA\_OWNERSHIP}\\
&\qquad\quad + \text{PUBLIC\_KEY\_REPLY})\\
&\qquad + (\text{Back off} * \text{PUBLIC\_KEY\_REQUEST}\\
&\qquad\quad + \text{SECURE\_TRANSACTION\_MESSAGE} + \text{ACK})].
\end{aligned} \tag{10}$$

The average transaction time is 57% less than the average delay experienced by a node as illustrated in Figure 4, where the average delay is computed from the values of minimum MAC delay (successful node) and maximum MAC delay (unsuccessful contending node) as shown in Figure 3. Irrespective of the other control/management packets (messages) used in RASCAAL, the delay value is computed only for the data packet which is transmitted in the SECURE_TRANSACTION_MESSAGE. This higher value of delay results because of the higher delay experienced by the contending node (Figure 3).

The salient feature of RASCAAL's security is the very short life span of a randomly formed dynamic cluster, which is equivalent to the duration of the current transaction. This short existence and randomness in cluster formation with no prior knowledge of the cluster head makes the system robust and difficult to attack. RASCAAL will have numerous short-lived random clusters formed in the entire life span of the ad hoc network. In RASCAAL, the ACTIVE CA node ensures the overall security and verifies the public keys of the communicating nodes, thus any malicious node trying to impersonate or track the role of an ACTIVE CA node will not be successful because the transaction time on average is 57% less than the average delay value experienced by a node. Therefore a malicious node will not be able to detect, transmit or send information on the channel before the role of the ACTIVE CA is randomly shifted to another unknown CA to form a new random short-lived cluster. The delay values in Figure 4 reach steady state since the randomly generated uniform deviate also reaches steady state.

4.2. Node Access Delay (Communicational + Computational). Simulations were performed in order to analyse the effects of MAC-related communicational and cryptographic-related computational delay on the performance of RASCAAL. The delay values were computed for a data packet sent in the SECURE_TRANSACTION_MESSAGE. Different encryption and hash algorithms (as outlined in Table 3) that could be used to provide the cryptographic needs of the RASCAAL protocol were considered in the simulations. Figure 5 shows both the communicational delay and the computational delay experienced by a data packet when the SHA-256 hash function is considered with different encryption algorithms. It is evident from Figure 5, that the security provided by RASCAAL is achieved at the cost of high communicational delay (average 6.6 seconds, 99.8% of total delay). This delay value could be avoided if the SECURE_TRANSACTION_MESSAGE was transmitted without any prior security-related control/management messages such as the ACTIVE_CA_MESSAGE. However, any security/authentication protocol will incur a certain value of MAC-related communicational delay based on the nature of such protocols. In terms of computational delay, the elliptic curve-based algorithms produce the largest delays while the delay associated with the Rabin, NtruEncrypt and RSA algorithms is negligible. The communicational and computational delay values for three different hash functions are outlined in Figure 6. When communicational delay is considered, there is little difference between the three hash function selections.

4.3. Energy Consumption (Communicational + Computational). RASCAAL was also analyzed in relation to its communicational and computational energy consumption. As described in Section 3, the communicational energy consumption was calculated by assuming that a station's operating current was 290 mA for transmitting, 205 mA for receiving and 62 mA for doze or sleep mode (Prism 2.5, 3.3 volts, IEEE 802.11b network interface card). Figure 7(b) shows the total network energy consumption (average communicational plus computational) when the SHA-256 hash function is considered with different encryption algorithms. Since the computational energy consumption values are too small to be visible in Figure 7(b), these are shown separately in Figure 7(a). It is clear that the communicational energy consumption is significantly greater than the computational energy consumption, accounting for 99.9% of the total energy. Figure 8 also outlines the computational only energy consumption associated with the different encryption algorithms when the SHA-256 hash function is employed. The RSA algorithm architecture consumes the most energy of the algorithms considered while the NtruEncrypt design consumes the least.

Figure 8 illustrates the communicational and computational energy consumption for the three different hash function designs studied. As with power consumption, the contribution of the hash functions is negligible with respect to the total energy consumption.

4.4. Saving in Energy Consumption (Communicational). The standard CSMA/CA protocol requires the contending nodes to always listen to the channel and stay awake for a duration equivalent to the value of the network allocation vector (NAV). This results in additional communicational energy consumption, which is included in the graphs previously outlined. The average value of energy consumed by the different encryption algorithms when the SHA-256 hash function is utilised, is 0.817 Joules, as shown in Figure 7. If the contending nodes change from an active state of continuously listening to the channel, to a doze or sleep state equivalent to the duration of NAV, the communicational energy consumed is reduced to 0.733 Joules, which results in an average saving of 10.3% and a maximum saving of 12.5%.

4.5. RASCAAL Overall Performance Discussion. From the analysis presented in this paper, it is clear that RASCAAL's performance suffers principally from communicational-related delay and energy consumption. However, all authentication protocols proposed for ad hoc networks will incur these MAC-related overheads which are associated with the protocol's control and management messages. In RASCAAL, there are only three management/control messages prior to a data packet being transmitted between two nodes. However, other authentication schemes that have been proposed for wireless ad hoc networks incur much greater communicational overheads. In threshold cryptography the nodes wishing to communicate have to firstly transmit the partial certificate collection requests to the server nodes obeying all the rules of the underlying MAC. Nodes must then wait and keep listening to the channel with their receivers on to receive the replies from the server nodes. It is clear that this will result in very high delay and energy consumption values. Additionally the threshold scheme is highly dependent on routing algorithms to find and route the certificate collection requests to the server nodes holding a share of the system secret. ID-based cryptography requires nodes to execute a number of algorithms such as setup, encrypt, extract and execute before they can actually start communication which will also inevitably result in significant communicational and computational delay and energy consumption losses.

RASCAAL's computational delay and energy consumption will be affected by the encryption algorithm and hash function architectures chosen to perform the cryptographic requirements of the protocol. Since the nodes in wireless ad hoc networks are typically resource constrained, the area of the cryptographic algorithm architecture must be considered in addition to their delay and energy consumption values. The cryptographic designs chosen for this investigation were selected as they were targeted at resource constrained applications. It is very difficult to compare these architectures as they are implemented on different technologies and since they use various key sizes, they offer different security strengths. However, their purpose is to provide an indication of the computational delay that is to be expected when cryptographic algorithms, such as those chosen for this study, are utilized in conjunction with security protocols. With respect to node delay, the elliptic curve-based cryptographic encryption algorithm designs are slower in comparison to the Rabin, NtruEncrypt and RSA architectures. However when energy consumption is also considered the RSA design performs poorly. When the area of the different architectures is taken into account, the area of the Rabin design is 3 times that of the elliptic-based designs. Therefore, overall the NtruEncrypt and the ECC algorithm design appear to be the most appropriate encryption functions to use with the RASCAAL protocol. The effect of different hash function architectures on the computational delay and energy consumption figures is almost negligible. However, when the design area is considered the SHA-1 architecture has the lowest gate count by approximately 50%. Therefore, of the hash functions studied, the SHA-1 design would be the most appropriate for use with RASCAAL.

5. Conclusion

All security or authentication protocols developed for wireless ad hoc networks have to obey the rules of the underlying IEEE 802.11 MAC to transmit security-related messages while still maintaining a certain quality of service. However MAC-related communicational effects and the computational effects of the cryptographic algorithms employed by protocols significantly affect the performance of protocols defined at layer two or above for the provision of security in ad hoc networks. This paper describes a novel authentication protocol, RASCAAL. Its performance is analysed by taking into account the effects of MAC-related communicational and cryptographic-related computational losses. RASCAAL is the first authentication protocol which proposes the concept of dynamically formed short-lived random clusters with no prior knowledge of the cluster head. To achieve this, RASCAAL implements the idea of a random ACTIVE CA selection and CA role shift in the network by integration with the underlying MAC for ad hoc networks. The performance analysis demonstrates that MAC related communicational losses contribute significantly to the total losses incurred by RASCAAL, in comparison to the cryptographic related computational losses. This research illustrates that research into security protocols for wireless ad hoc networks needs to be considered in the context of the overall system. However, it was found that communicational energy saving of up to 12.5% can be achieved by changing the status of the wireless nodes from a receive state to a sleep or doze state during the course of operation. In determining the cryptographic-related effects, various encryption and hash function hardware designs that have been proposed in literature for use in resource constrained applications, and which could be used with the RASCAAL protocol, were studied. Their purpose was to provide an indication of the computational effects that can be expected when cryptographic algorithms are utilized in conjunction with security protocols. From this study, it was found that the

NtruEncrypt and ECC encryption algorithms and SHA-1 hash function were particularly suitable for employment with the RASCAAL protocol.

Future work will involve analysing RASCAAL for an increased number of wireless CA and non-CA nodes bringing more randomness to the selection of the ACTIVE CA node. This increase in the density of nodes will require an investigation into suitable routing protocols that can provide multi-hop operation between nodes. An implementation of RASCAAL on top of energy constrained MAC for sensor networks will also be carried out and a comparison conducted between the performance of RASCAAL for sensor and ad hoc networks.

Acknowledgment

The authors wish to thank EPSRC (EP/C006976/1) for funding this research.

References

[1] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing robust and ubiquitous security support for mobile ad-hoc networks," in *Proceedings of the International Conference on Network Protocols (ICNP '01)*, pp. 251–260, 2001.

[2] J. S. Stach, E. K. Park, and Z. Su, "An enhanced authentication protocol for personal communication systems," in *Proceedings of the IEEE Workshop on Application-Specific Software Engineering and Technology (ASSET '98)*, pp. 128–132, 1998.

[3] A. Aziz and W. Diffie, "Privacy and authentication for wireless local area networks," *IEEE Personal Communications*, vol. 1, no. 1, pp. 25–31, 1994.

[4] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[5] L. Zhou and Z. J. Haas, "Securing ad hoc networks," *IEEE Network*, vol. 13, no. 6, pp. 24–30, 1999.

[6] H. Lou, P. Zerfos, J. Kong, S. Lu, and L. Zhang, "Self securing ad hoc wireless networks," in *Proceedings of the 7th IEEE International Symposium on Computers and Communications (ISCC '02)*, pp. 567–574, 2002.

[7] B. Lehane, L. Doyle, and D. O'Mahony, "Shared RSA key generation in a mobile ad hoc network," in *Proceedings of the IEEE Military Communications Conference (MILCOM '03)*, vol. 2, pp. 814–819, 2003.

[8] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of the Conference on Advances in Cryptology (CRYPTO '84)*, vol. 7 of *Lecture Notes in Computer Science*, pp. 47–53, 1984.

[9] H. Deng, A. Mukherjee, and D. P. Agrawal, "Threshold and identity-based key management and authentication for wireless ad hoc networks," in *Proceedings of the IEEE International Conference on Information Technology: Coding Computing (ITCC '04)*, vol. 1, pp. 107–111, 2004.

[10] D. Bonh and M. Franklin, "Identity based encryption from Weil pairing," in *Advances in Cryptology*, vol. 2139 of *Lecture notes in Computer Science*, pp. 213–229, Springer, New York, NY, USA, 2001.

[11] R. Li, J. Li, H. Kameda, and P. Liu, "Localized public-key management for mobile ad hoc networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '04)*, vol. 2, pp. 1284–1289, 2004.

[12] M. Bechler, H.-J. Hof, D. Kraft, F. Pahlke, and L. Wolf, "A cluster-based security architecture for ad hoc networks," in *Proceedings of the 23rd IEEE Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 4, pp. 2393–2403, 2004.

[13] Q. Ren and Q. Liang, "Secure media access control (MAC) in wireless sensor networks: intrusion detections and countermeasures," in *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '04)*, vol. 4, pp. 3025–3029, 2004.

[14] S. Zhu, S. Xu, S. Setia, and S. Jajodia, "LHAP: a lightweight hop-by-hop authentication protocol for ad-hoc networks," in *Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops (ICDCSW '03)*, pp. 749–755, 2003.

[15] G. A. Safdar and M. McLoone, "Randomly shifted certification authority authentication protocol for MANETs," in *Proceedings of the 16th IST Mobile and Wireless Communications Summit*, pp. 1–5, 2007.

[16] ANSI/IEEE STD 802.11, "IEEE 802.11b, Part II, Wireless LAN Medium Access Control and Physical Layer Specifications," 1999.

[17] OPNET, "Modeller Documentation-Wireless Module User Guide," http://www.opnet.com.

[18] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Transactions on Computer Systems*, vol. 8, pp. 18–36, 1990.

[19] G. Gaubatz, J.-P. Kaps, and B. Sunar, "Public key cryptography in sensor networks-revisited," in *Proceedings of the 1st European Workshop on Security in Ad Hoc and Sensor Networks (ESAS '05)*, pp. 2–18, 2004.

[20] L. Batina, N. Mentens, K. Sakiyama, B. Preneel, and I. Verbauwhede, "Public-key cryptography on the top of a needle," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '07)*, pp. 1831–1834, 2007.

[21] M. Feldhofer and J. Wolkerstorfer, "Strong crypto for RFID tags—a comparison of low-power hardware implementations," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '07)*, pp. 1839–1842, 2007.

[22] C. Yeh, E. F. Hsu, K. W. Cheng, J. S. Wang, and N. J. Chang, "An 830 mW, 586 kbps 1024 bit RSA chip design," in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE '06)*, pp. 24–29, 2006.

[23] J.-P. Kaps and B. Sunar, "Energy comparison of AES and SHA-1 for ubiquitous computing," in *Emerging Directions in Embedded and Ubiquitous Computing*, X. Zhou, et al., Ed., vol. 4097 of *Lecture Notes in Computer Science*, pp. 372–381, Springer, Seoul, Korea, August 2006.

[24] IEEE Std 1363.1-2008, "IEEE Standard specification for public key cryptographic techniques based on hard problems over lattices," pp. C1–C69, 2009.