Identity Establishment and Capability Based Access Control (IECAC) Scheme for Internet of Things

Mahalle, Parikshit N.; Anggorojati, Bayu; Prasad, Neeli R.; Prasad, Ramjee

Published in:
The 15th International Symposium on Wireless Personal Multimedia Communications

Publication date:
2012

Document Version
Early version, also known as pre-print

Identity Establishment and Capability Based Access Control (IECAC) Scheme for Internet of Things

Parikshit N. Mahalle, Bayu Anggorojati, Neeli Rashmi Prasad and Ramjee Prasad
Center for TeleInFrastruktur (CTIF) – Aalborg University – Denmark
Email :{ pnm, ba, np, prasad} @es.aau.dk

Abstract— **Internet of Things (IoT) become discretionary part of everyday life and could befall a threat if security is not considered before deployment. Authentication and access control in IoT is equally important to establish secure communication between devices. To protect IoT from man in middle, replay and denial of service attacks, the concept of capability for access control is introduced. This paper presents Identity establishment and capability based access control (IECAC) protocol using ECC (Elliptical Curve Cryptography) for IoT along with protocol evaluation, which protect against the aforementioned attacks. The protocol evaluation by using security protocol verification tool shows that IECAC is secure against these attacks. This paper also discusses performance analysis of the protocol in terms of computational time and compared with other existing solutions.**

Keywords-Access Control ; Capability ; Identity Establishment

I. INTRODUCTION

IoT is mandatory subset of future Internet where every virtual or physical thing can communicate with every other thing giving seamless service to all stakeholders. IoT is convergence of resource constrained sensors, RFID, smart devices and anything with sensing, computing and communication capability and the realistic notion of IoT [1] has been seen with the development of wireless communication and Internet access between these devices. Seamless communication between ubiquitous things in IoT possesses problems of authentication and access control. The greater scale and scope of IoT increases the options in which a user can interact with the things in his/her physical and virtual environment. IoT could be both distributed and ad-hoc in nature and therefore security problem are daunting. In multi-hop network, each node acts as a receiver and transmitter; attacker can gain access to resources and devices in the absence of authentication mechanism. Dynamic network topology due to mobile nodes, lower bandwidth than traditional network and energy constraints are another threats to IoT networks causing attacks like denial of service attack. Notion of identity establishment and authentication are closely related to each other. Identity establishment is process of associating user with another legitimate user or resource. Authentication is secure identification of entities in which proof of possession of an identity is verified. Devices ranging from sensors to RFID tags, identities extended to heterogeneous devices are the main challenges of IoT to devices, ubiquitous interaction and large numbers of design security solutions.

This paper is organized as follows. Section II evaluates and summarizes related work in the authentication in IoT. Section III presents proposed protocol for mutual identity establishment and access control. Section IV presents security analysis of different attacks and protocol verification. Section V presents protocol evaluation in terms of computational time and comparison with the other existing solutions. Section VI concludes the paper with future plans.

II. RELATED WORK

There is closely related work done in [2] where security association takes place with increased communication overhead and authentication is left unaddressed. Authors presented distributed access control solution based on security profiles but attack resistance is not explored. In [3], authors have presented ECC based authentication protocol but the major disadvantage is that they are not Denial of Service (DoS) attack resistant, which is of paramount important in IoT that consists of billions of devices. In [4], author addresses the problem of secure communication and authentication based on shared key and is applicable to limited location and cannot be used for wide area. It addresses the peer-to-peer authentication but cannot be extended in resource constrained environment. There has been lot of debate about which of the cryptographic primitives like PKI or symmetric crypto is suitable for the IoT. Most of the research has mainly focused in the area like Wireless Sensor Network (WSN) and its application. Many security mechanisms have been proposed based on the private key cryptographic primitives due to fast computation and energy efficiency. Scalability problem and memory requirement to store keys makes it inefficient to heterogeneous devices in IoT. Public key cryptography based solution overcomes these challenges with high scalability, low memory requirements and no requirement of key pre-distribution infrastructure. In [5], authors have presented ECC based mutual authentication protocol for IoT using hash functions. Mutual authentication is achieved between terminal node and platform using secret key cryptosystem introducing the problem of key management and storage. Self- certified keys cryptosystem based distributed user authentication scheme for WSN is presented in [6] where only user nodes are authenticated and is not lightweight solution for IoT. In [7], author presents authentication with Parameter passing during handshake.

TABLE 1. STATE OF THE ART EVALUATION SUMMARY

Solutions	Parameters						
	Mutual Authentication	*Lightweight Solution*	*Attack Resistant*			*Distributed Nature*	*Access Control*
			DoS	*Man in Middle*	*Replay*		
[2]	No	No	No	No	No	Yes	Yes
[3]	Yes	Yes	No	No	No	Yes	No
[4]	No	No	Yes	Yes	Yes	No	No
[5]	Yes	Yes	No	Yes	Yes	Yes	No
[6]	No	No	No	No	No	Yes	No
[7]	Yes	Yes	No	Yes	Yes	No	No
[8]	Yes	No	No	No	No	Yes	No
[9]	Yes	No	No	No	No	Yes	No

[2]: Ubiquitous access control in MAGNET , [3]: ECC based authentication in RIFID , [4]: Authentication Ad-hoc wireless network , [5]: Authentication in IoT , [6]: Authentication in WSN , [7]: Progressive authentication in Ad-hoc Network , [8]: Peer identification and authentication , [9]: Authentication in Ad-hoc network

Handshake process is time consuming and also based on symmetric key cryptography that consumes more memory for large prime numbers. Efficient identification and authentication presented in [8] and is based on the signal properties of node but is not suited for mobile nodes. Direction of the signal is considered as parameter for node authentication but it takes more time to decide signal direction with more memory and computations involved. In [9], cluster based authentication is proposed which is most suited for futuristic IoT, but attacker can get hold of distribution of system key pairs and cluster key. Generation of random numbers and signatures creates considerable computational overhead consuming memory resources.

State of the art evaluation is shown in Table 1. Related work is summarized based on the parameters like mutual authentication, lightweight solution, resistant to attacks, distributed nature and access control solution. From table 1, it is clear that, all existing solutions for authentication and access control do not fulfill all requirements for IoT. Objective is to achieve mutual identity establishment i.e. authentication and once authenticated, access control will take place. Paper proposes new method of authentication of devices and access control for the IoT using public key approach with scalability and less memory requirements. Most important design issue of IoT is the mobility of heterogeneous devices and our scheme works efficiently for this need.

III. PROPOSED IECAC SCHEME IN IOT

Algorithm presented in this paper addresses both authentication and access control which are divided into three parts:

A. Secret key generation based on Elliptical Curve Cryptography-Diffie Hellman algorithm (ECCDH)
B. Identity Establishment
C. Capability creation for access control

A. *Secret key generation based on ECCDH and identity establishment for authentication*

There is considerable interest in ECC for IoT security [10].It has advantages of small key size and low computation overhead. It uses public key cryptography approach based on elliptic curve on finite fields. ECCDH [10] is a symmetric key agreement protocol that allows two devices that have no prior knowledge about each other to establish a shared secret key which can be used in any security algorithm. Using this public parameter and own private parameter, these parties can calculate the shared secret. Any third party, who doesn't have access to the private details of each device, cannot calculate the shared secret from available public information. All devices joining IoT shares key pair during the bootstrapping. IECAC scheme presented in this paper is also applicable to security bootstrapping. Security bootstrapping is the process by which devices join the IoT with respect to location and time. It includes device authentication along with credential transfer. Protocol uses one or more trusted Key Distribution Center (KDC) to generate domain parameter and other security material and important part is this KDC is not required to be online always. Initially KDC randomly selects particular elliptic curve over finite field *GF (p)* where *p* is a prime and makes base point *P* with large order *q* (where q is also prime). KDC then picks random $x \varepsilon GF(p)$ as a private key and publishes corresponding public key $Q = x \times P$. KDC generates random number $Ki \varepsilon GF(p)$ as a private key for device $_i$ and generates corresponding public key $Q_i = Ki \times P$. The key pair $\{Q_i , Ki\}$ is given to device i. With the increasing number of devices, KDC can generate ECC key pair based on base point P for any number of devices as it is rich is terms of resources as compared to other devices in IoT. These ECC key pairs will be used to share common secret key for secure communication using ECCDH and is explained below. Steps of aforementioned ECCDH are shown presented in Fig. 1. Assumption is that ECC is running at trusted KDC. There is an agreement on system based point P and generate *(Qu , Ku)* and *(Qh , Kh)* pairs where

Qu = Public key of Device 1
Ku = Secret key of Device 1
Qh = Public key of Device 2
Kh = Secret key of Device 2

And *P* is large prime number over *GF (P)* and generations of above keys are as follows:

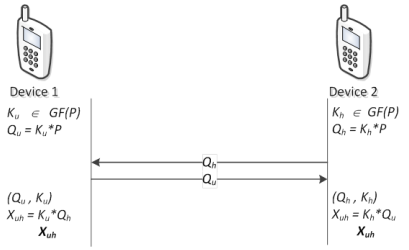

Fig. 1. ECCDH for Establishing Shared Secret Key

No parameter is disclosed in this process of establishing a shared secret key other than domain parameter *P* and public keys. This paper consider sensor node as device, because the functionalities and operational principle of wireless sensor networks makes it appropriate and mandatory candidate of the IoT.

B. Protocol for Identity Establishment

1) One way authentication: One way authentication authenticates Device 1 to Device 2 and is explained below. As per above ECCDH, both Device 1 and Device 2 has Xuh as a common secret key. Device 1 selects r ∈ GF (P) which will be used to create session key. Tu is generated as a time stamp by Device 1. It is assumed that synchronisation is taken care using appropriate mechanism. Secret key is created by Device 1 as $L = h (X_{uh} \oplus Tu)$. Then , Device 1 encrypts r with secret key L as $R = E_L (r)$ and encrypts Tu by Xuh as $Tus = E_{Xuh} (Tu)$. After this Device 1 builds a Message Authentication Code (MAC) value as $MAC_1 = MAC(X_{uh} , R \| ICAP_1)$ where $ICAP_1$ is a data structure representing an identity based capability for this Device 1 giving access rigts. Details about ICAP are given in the same section below. Now Device 1 sends following parameters to Device 2 directly or through gateway node / coordination node or access point as (R, T_{us}, MAC_1). Device 2 generates it's current time stamp as $T_{current}$ and Device 2 will decrypt Tus to get T_u and compare it with $T_{current}$. If $T_{current} > T_u$, it is valid.

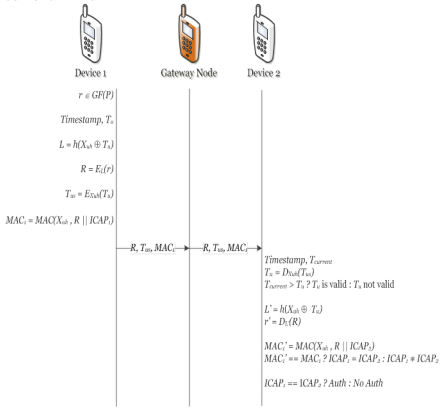

Fig. 2. One Way Authentication Protocol

Now Device 2 calculates *L* and decrypt *R* to get *r*. Device 2 also calculates the MAC_1 ' and it will verify this with MAC_1 received from Device 1. If valid, then Device 1 is authentic to Device 2. Device 1 also matches the *ICAP1* received with *ICAP2* stored at Device 2. If Device 2 gets match with *R* , MAC_1 , Tus then Device 1 is authenticated to Device 2. Aforementioned protocol is presented in figure 2 given below.

2) Mutual authentication: This part of authentication authenticates Device 2 to Device 1, and is explained below in figure 4. Device 2 builds a MAC as $MAC_2 = MAC (r \| ICAP_2)$ and also encrypts r with X_{uh} as $R' = E_{Xuh} (r)$. Device 2 sends (R' , MAC_2) to Device 1. Device 1 verifies MAC_2 and decrypt R' and compare received r with this r (denoted as r' and r'' in figure) . If match found , Device 2 is also authenticated to Device 1 and communication and access will be granted based on the $ICAP_2$. This protocol achieves both mutual authentication along with capability based access control in secure way.

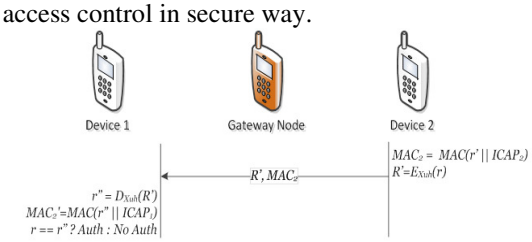

Fig. 3. Protocol for Mutual Authentication

3) Capability access for access control: Conceptually, a capability is a token that gives permission to access device. A capability is implemented as a data structure that contains two items of information: a unique device identifier and access rights. For simplicity, it is sufficient to examine the case where a capability describes a set of access rights for the device. Device which may also contain security attributes such as access rights or other access control information. The ICAP (Identity based Capability) [11] was essentially extending the Capability system concept, in which the capability is used by any User or Subject that wants to get access to a certain device or Resource.

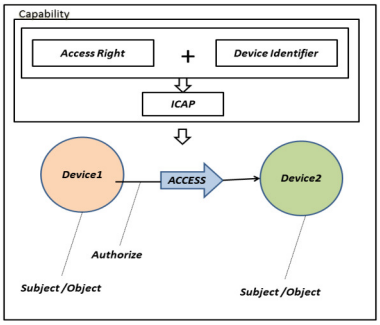

Fig. 4. Capability structure

If the capability that is presented by the Subject matches with the capability that is stored in the device or an entity that manages the device, access is granted. However, unlike the classical capability based system, ICAP introduced the identity of Subject or User in its operation. In this way, it claimed to reduce the number of capabilities stored in the so-called "Object Server" or "Gateway" or "Access Point" and thus offers more scalability. Moreover, it has better control in capability propagation which provides more efficient access later on. ICAP structure is shown in 4 with how capability is used for access control. ICAP is represented as

$$ICAP = (ID, AR, Rnd)$$

Where:

- *ID*: Device identifier
- *AR*: Set of access rights for the device with device identifier as *ID*
- *Rnd*: Random number to prevent forgery and is a result of one way hash function as: $Rnd = f (ID, AR)$

In IECAC, access rights are sent in the form of MAC value in the authentication process.

IV. EVALUATION AND ANALYSIS

The evaluation will focus on identity establishment in terms of one way and mutual as the most important processes in the authentication. The Automated Validation of Internet Security Protocols and Applications (AVISPA) tool [12] based on Dolev-Yao model [13] is used for model and protocol verification. We implement aforementioned protocol in the stages. First stage of protocol authenticates Device 1 to Device 2 and i.e. one way authentication and second stage of protocol is for mutual authentication i.e. authenticates Device 2 to Device 1. Verification results are described below.

A. *Evaluation procedure*

In order to carry out the evaluation using AVISPA some assumptions are made. Both the devices have already obtained ECC based shared key using Diffie-Hellman (ECCDH). As stated earlier, assumption here is that KDC is secure and trusted. Complete protocol evaluation is presented in following model:

$D1 \rightarrow D2:[R,T_{us},MAC_1];[\{r\}\_L,\{T_u\}\_X_{uh},RND_1]$

$D1 \leftarrow D2:[R',MAC_2];[\{r\}\_X_{uh},RND_2]$

Where:

- *D1*: Device 1
- *D2*: Device 2
- { } _: A symbol of encryption
- T_u : Timestamp generated as a nonce
- X_{uh} : A shared key between *D1* and *D2* using ECCDH
- r : Some value $x \in GF(p)$
- RND_1 : *MAC* value of X_{uh}, R and $ICAP_1$ where *ICAP* is result of one way hash function *f(Device_ID, Access Rights, Rnd)*, *Rnd* is random number generated to prevent forgery
- RND_2: *MAC* value of r and $ICAP_2$
- L : result of one way hash function (*XOR* of X_{uh} and T_u)

Besides this, Dolev-Yao intruder model has been introduced in the evaluation. The intruder is assumed to have the knowledge of the following:

- *ID*: Device identifier
- $f()$: Knowledge of one way hash function

B. *Evaluation results*

The goal of evaluation is to verify protocol for attacks mentioned above and ensures mutual authentication along with access control.

1) Mutual authentication: X_{uh} is shared securely between *D1* and *D2* and *r* is provided by trusted KDC to both the devices. Consequently, *D1* is authenticated to *D2* as only D2 can decrypt *R* and T_{us}. Also *MAC* can be calculated only by *D2* and *D2* is sending encrypted *r* to authenticate it to *D1*. Verification results show that secure mutual authentication is achieved.

2) Man in middle attack: In case of authentication, even there is man in middle attack on *R*, T_{us}, MAC_1 parameters; attacker will not reveal any information. AVISPA shows that authentication protocol is free from attacks. For access control, man in the middle attacks happen when an attacker eavesdrop the *ID* and *ICAP* transmitted, and then masquerade attack happens when the attacker uses the stolen *ID* and *CAP*. The key to preventing masquerade attack from the stolen *CAP* is to use *ID* to validate the correct device. If the attacker manages to steal the *ID*, the attack is prevented by applying public key cryptography to *ID*, assuming that the authentication process has been done before access control. In this way, although the attacker gets the *ICAP* which is not encrypted, the capability validity check will return an exception because the one way hash function, *f(ID, AR, Rnd)* will return a different result than the one presented in the *CAP*, without a correct *ID*.

Another type of man-in-middle attack is replay attack. Adversary can intercept the message sent out from *D1*. However, it is not possible in IECAC because it can easily detect by verifying timestamp T_u. If T_u is older than predefined threshold value, it is invalid and has been used. If T_u is changed, $MAC_1 = MAC(X_{uh}, R \,||\, ICAP_1)$ is not valid and consistent. For access control, IECAC prevents the replay attack by maintaining the freshness of Rnd, for example by using time stamp or nonce by including *MAC* as well. Even if the attacker manages to compromise the solution and gets the *ICAP*, it cannot use the same capability next time because the validity will be expired.

3) DoS attack: Upon receiving the message from *D1*, *D2* first checks the validity of timestamp. If it is not valid, then *D2* discards the message. Otherwise, it computes a MAC_2 value to compare with received value. DoS happens when an attacker accesses a particular resource massively and simultaneously by using the same or different *ID*s. It is easy to control access using one *ID* because the system is able to maintain the session, thus the access of the same *ID* to the same resource can be restricted to only one session at a time. The potential of DoS attacks from multiple *ID*s can be prevented in the capability propagation process. Therefore, DoS attack can be prevented or at least minimized.

V. PERFORMANCE EVALUATION

Security level of protocol presented in this paper depends on the type of MAC algorithm, encryption algorithm and security level of ECC signature. We propose to use RC5 stream cipher for encryption, which takes 0.26 ms on Mica2 motes [14, 15 and 16]. RC5 is notable for its simplicity for resource constrained devices such as IoT and its flexibility due to the built in variability. Heavy use of data independent rotations and mixture of different operations provides strong security to RC5 [17].

We propose to use SHA-1 as one way hash function which takes 3.63 ms on Mica2 motes and it is computationally expensive to find text which matches given hash and also it is difficult to two different texts which produces the same hash [14, 15, and 16]. To generate the MAC value, we propose CBC-MAC which has advantage of small key size and small number of block cipher invocations and takes 3.12 ms on Mica2 motes [15].The time required to generate random number is 0.44 ms and ECC to perform point multiplication which takes 800 ms on Mica2 motes

[15,16]. In IECAC protocol as the message length is fixed, CBC-MAC is most secure [18]. It is clear from these values that maximum time is required for ECC point multiplication. In IECAC, point multiplication is taking place at KDC and as KDC is powerful device, computational overhead is trivial as compared to the sensors. We denote the computational time required for each operation by device in IoT by following notation:

D_H = Time to perform one way hash function SHA-1
D_{MAC} = Time to generate Mac value by CBC-MAC
D_{RC5} = Time to perform encryption and decryption by RC5
D_{MUL} = Time to perform ECC point multiplication
R = Time for random number generation

TABLE 2. COMPUTATIONAL TIME FOR IECAC

Scheme	IECAC	HBQ [19]	IoT_Auth [5]
Auth. Time	$2D_H + 2D_{MAC} + 2D_{RC5}$	$2D_H + 2D_{MAC} + D_{RC5} + 3D_{MUL}$	$R + D_H + 2D_{MUL}$
Total	$2D_H + 2D_{MAC} + 2D_{RC5}$	$2D_H + 2D_{MAC} + D_{RC5} + 3D_{MUL}$	$R + D_H + 2D_{MUL}$
Total time	14.02 ms	2413.76ms	1604.07ms

Table 2 shows the comparison of computational time for above-mentioned protocol. IECAC protocol for mutual authentication and access control for the IoT devices takes less time (14.28 ms) as compared to other protocol compared in this paper. Key point to note here is that, none of the work has addressed issue of authentication and access control as an integrated solution for IoT. Total computational time for of the proposed scheme, HBQ [19] and mutual authentication for IoT (IoT_Auth) [5] is shown in table 2. IoT_Auth scheme requires $R + D_H + 2D_{MUL}$ time for mutual authentication which comes approximately 1604.07 ms. HBQ scheme takes $2D_H + 2D_{MAC} + D_{RC5} + 3D_{MUL}$ total time for authentication which is approximately 2,413.76 ms. Key point to note here is that both the schemes do not address access control after authentication. IECAC takes only $D_H + 2D_{MAC} + 2D_{RC5}$ which takes only 14.02 ms which is much better than other two schemes analyzed in this paper. In IECAC, $2D_H$ factor is introduced which comprises time required by one way hash function in authentication as well as in *ICAP* to calculate *Rnd*.

VI. CONCLUSION AND FUTURE WORK

Distributed, lightweight and attack resistant solutions, being the most favorable choices for IoT , puts resilient challenges for authentication and access control of devices. Paper presented efficient and scalable ECC based authentication and access control protocol. Protocol is divided in two phases as one way authentication and mutual authentication and integrated with capability based access control solution. Power of ECC is extended to achieve mutual authentication of devices with novel capability based approach for access control.

Furthermore, paper presents comparative analysis of different authentication and access control schemes for IoT. Comparison in terms of computational time shows that IECAC scheme is efficient as compared to other solution. Protocol is also analyzed for the performance and security point of view for different possible attacks in IoT scenario. Protocol evaluation shows that it can defy attacks like DoS, man-in-middle and replay attacks efficiently and effectively. Paper also presents protocol verification using AVISPA tool which proves that the IECAC protocol is also efficient for large scale devices in terms of key sharing and authentication. Future plan is to put this protocol in place with RFID middleware architecture for Identity management in IoT.

REFERENCES

[1] M. Weiser, “The computer for the 21st century,” Scientific American, vol. 265, pp. 66-75, 1991of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, Apr. 1955.

[2] D. M. Kyriazanos, G. I. Stassinopoulos and N. R. Prasad, "Ubiquitous Access Control and Policy Management in Personal Networks," Mobile and Ubiquitous Systems: Networking & Services, 2006 Third Annual International Conference on , vol., no., pp.1-6, July 2006.

[3] S. I. Ahmed, F. Rahman and E. Hoque, “ERAP: ECC based RFID Authentication Protocol,” 12th IEEE International Workshop on Future Trends of Distributed Computing Systems, 2008.

[4] D. Balfanz, D. K. Smetters, P. Stewart and H. C. Wong, “Talking to strangers: Authentication in ad-hoc wireless networks”, Network and Distributed Systems Security Symposium (NDSS), Feb 2002.

[5] G. Zhao, X. Si, J. Wang, X. Long and T. Hu, "A novel mutual authentication scheme for Internet of Things," Modelling, Identification and Control (ICMIC), Proceedings of 2011 IEEE International Conference on , vol., no., pp.563-566, 26-29 June 2011

[6] C. Jiang, B. Li and H. Xu, “An efficient scheme for user authentication in wireless sensor networks,” in: 21st International Conference on Advanced Information Networking and Applications Workshops, 2007, pp.438-442.

[7] R.R.S. Verma , D.O'Mahony and H.Tewari , "Progressive authentication in ad hoc networks. In Proceesings of the Fifthe European Wireless Conference, February 2004.

[8] T. Suen, A.. Yasinsac, "Ad hoc network security: peer identification and authentication using signal properties," Information Assurance Workshop, 2005. IAW '05. Proceedings from the Sixth Annual IEEE SMC , vol., no., pp. 432- 433, 15-17 June 2005.

[9] L. Venkatraman, D.P. Agrawal, "A novel authentication scheme for ad hoc networks," Wireless Communications and Networking Conference, 2000. WCNC. 2000 IEEE , vol.3, no., pp.1268-1273 vol.3, 2000.

[10] N. Koblitz, Elliptic curve cryptosystems, in Mathematics of Computation 48, 1987, pp. 203–209.

[11] L. Gong, “A Secure Identity-Based Capability System,” IEEE Symposium on Security and Privacy, 1989.

[12] Avispa – a tool for Automated Validation of Internet Security Protocols. http://www.avispa-project.org.

[13] D. Dolev and A. C.-C. Yao. On the security of public key protocols. In FOCS, pages 350–357. IEEE, 1981.

[14] R. Chakravorty, A Programmable Service Architecture for Mobile Medical Care. 4th IEEE International Conference on Pervasive Computing and Communications, 2006.

[15] C. Karlof, N. Sastry, and D. Wagner. Tinysec: link layer security architecture for wireless sensor networks. In SensSys, ACM Conference on Embedded Networked Sensor Systems, 2004.

[16] N. Gura, A. Patel, A. Wander, H. Eberle, S. C. Shantz,Comparing Elliptic Curve Cryptography and RSA on 8-it CPUs. In CHES 2004, Vol. 3156, LNCS, pp.119-132.

[17] Y.L. Yin, The RC5 encryption algorithm: two years on, *CryptoBytes* (3) 2 (Winter 1997).

[18] M. Bellare, J. Killan, and P.Rogaway,: The security of cipher block chaining,In:Desmedt, Y(ed.),CRYPTO 1994. LNCS, VOL.839, pp.341-358. Springer,Heidel-verg(1994)

[19] H. Wang, B. Sheng, Q. Li, Elliptic curve cryptography based access control in sensor networks, Int. J. Security and Networks, Vol. 1, Nos. 3/4, pp.127–137 (2006).