An efficient Authentication Scheme Using Token Distribution for Cloud-based Smart Home

Ruiming Kou
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China
kouruiming@iie.ac.cn

Yazhe Wang
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China
wangyazhe@iie.ac.cn

***Abstract*—Due to the exponential growth of wireless devices and the rapid development of smartphone's computing and storage capabilities, interests on using smartphone to remote access and manipulate smart home facilities have been increased greatly in recent days. However, this application context requires smartphones to store, process and transfer some security-critical data, which makes them vulnerable to various security threats. Therefore, for smart home application context, robust remote user authentication should be considered. This paper presents an efficient cloud-based communication architecture and a confidential authentication scheme that provide a guaranteed secure remote access to target smart home devices. The proposed scheme aims to satisfy several authentication requirements including flexibility, security and robustness by introducing a series of technologies such as cloud, electronic access token, ARM TrustZone. Our solution also allows users to delegate their access rights and revoke them either automatically or manually whenever necessary. Moreover, a security analysis is provided to demonstrate whether our scheme could defend the common security attacks.**

Keywords—Smart Home; Cloud; Mobile Security; ARM TrustZone; Access Control; Electronic Access Token; User Authentication

I. INTRODUCTION

Nowadays, due to increasing computing and storage capabilities of smart terminal platforms, smart home devices integrated with wireless module begin to enter the home of normal people and enormously change their living style. With the help of the Internet, people with Internet-connected smartphone gain the abilities to control as well as supervise their home facilities anytime and anywhere. In this scenario, cloud is well-suited for such applications for its high flexibility and low cost storage. Meanwhile, compare to the growing trend of dramatic investigation into smart home devices and infrastructure, few researchers and institutes have ever realized the importance of the security of smart home facilities. However, the significance of the security of home facilities is conspicuous since once the control of the victims' home facilities are stolen or snatched by adversaries, not only the safety of the victim's property and privacy but even the individual's life security will face great threats.

Password authentication is one of the simplest as well as the most convenient authentication mechanisms, but the benefit could not cover the security vulnerabilities invited by its implementation, such risks include eavesdropping attack and phishing attack. In order to resolve these potential threats, one-time password(OTP)authentication scheme is invented and gradually become one of the most popular forms of securing network authentication Vaidya et al. [2], but this method meanwhile causes higher computational overhead which will lead to noticeable delay considering the limited hardware resources of smart home devices. Ibriq et al. [3] introduced a hierarchical key management system which includes a central trust authority and a authentication ticket for each authenticated node in smart home environment. This scheme successfully settles the heavy overhead caused by the re-authentication requests but with rigor premise: the central trust authority must be absolutely secure and the smart home devices must be regularly distributed and which is scarcely possible in the domestic scenario. Dmitrienko et al. [1] presents a design and implementation of an access control system for NFC-enabled smartphones. They used electronic access token to authenticate users and implemented the remote token issuing or revocation. Access right delegation is also introduced in this scheme. However, the function of Near Field Communication(NFC) raises a high demand for specific kind of smartphones and the NFC, limited by short nominal communication range, is not suitable for remote interaction between users' smartphone and smart home devices.

Smartphones recently are confronted with numerous security challenges such as rootkits attack and malware attack according to Bickford et al.[4]. In order to deal with these severe security vulnerabilities, ARM TrustZone was proposed and now becomes available on many smartphones. TrustZone separates conventional hardware domain into "normal world" and "secure world". The trusted execution environment provides a secure and isolated computing and storage zone for sensitive data. Therefore, smart home authentication system with high data protection requirement can benefit from the assistance of TrustZone technology.

In this paper, we propose an efficient cloud-based communication architecture between user mobile equipment and smart home devices. An authentication protocol is also designed which ensures the security of the communication process between them. Cloud is regarded as the medium to complement the authentication process such as transfer access

tokens between smart phones and corresponding smart home devices. ARM TrustZone is utilized to protect the secure-critical data on users' smart phones.

Different from conventional authentication model, Cloud-based smart home authentication system could provide a variety of appealing new features and meet users' distinctive demand in different scenarios. This system does not require users to hold any physical auxiliaries except an electronic access control token stored in their smartphones. Moreover, instead of transferring access token through NFC or Blue-tooth which require users' physical proximity, with the help of cloud, this system allows users to obtain access right and control privileges remotely through Internet. In addition, electronic access control token could offer more flexible features such as access right remote distribution and revocation, user delegation and context-aware or time-limited access control policies.

II. MODEL ANALYSIS

In this section, we first introduce the model of cloud-based smart home authentication scheme and then a related adversary model is presented for the analysis of basic assumptions.

A. System Model

The model of cloud-based smart home authentication system is depicted in Fig.1. All roles include cloud C, a smart home device D, a device owner O and a device del.user(delegated user) U. The cloud C represents a third-party cloud service provider which act as the communication medium between users and their private smart home service. Moreover, cloud C provides a basic user authentication to promise all connected users are well-authorized. Smart home device D is the representation of all sorts of related devices belong to device owner O. Device owner O is an individual person who solely enjoy the ownership of smart home device D. The device del.user U is a person who is authorized by O to access or manipulate the device D temporarily. Access right is distributed through the process of issuing electronic access tokenT_u. We should notice that a device D can have multiple del.users but only belongs to one owner.

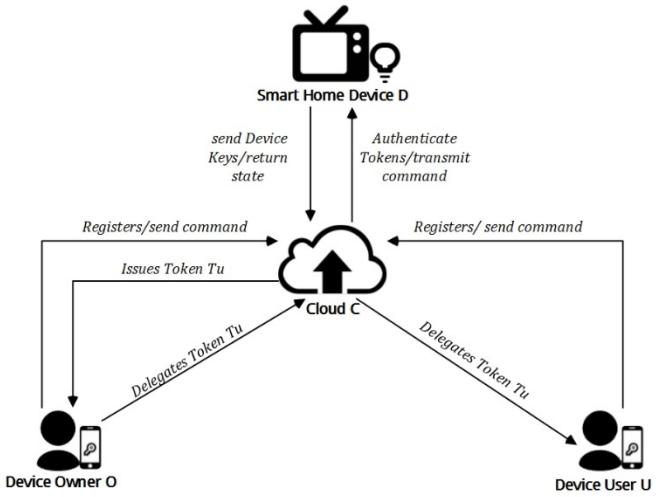

Fig.1. system architecture

B. Adversary Model

Communication channels. We assume that all the communication channels between each individual component are insecure. Potential adversary $\mathcal{A}$ could take control of all these channels so that $\mathcal{A}$ can eavesdrop, modify, insert, delete and re-route protocol messages that transmitted between D and C, C and U, O and C. Smartphone platform. ARM TrustZone separates the conventional smartphone computing and storage domain into "normal world" and "secure world". In this scenario, smartphone platform $\mathcal{P}$ consists of normal insecure environment $\mathcal{N}$ and a secure execution environment $\mathcal{S}$. We assume that $\mathcal{A}$ can only carry out malicious intentions such as malware attack in the range of $\mathcal{N}$. The computing process or storage of high sensitive data in $\mathcal{S}$, on the other hand, cannot be stolen, modified or compromised.

Smart home device. We assume that the smart home device D is trusted so that the data stored in D and related local data processing cannot be stolen, modified or compromised. Moreover, since most of smart home devices are installed in owners' private house, we assume $\mathcal{A}$ has no opportunity to approach those devices physically. Thus, $\mathcal{A}$cannot damage or reset target devices.

Cloud platform. We assume that Cloud C involved in our smart home authentication system is trusted. Related data and function flow cannot be controlled by $\mathcal{A}$. Hence, attacks against the cloud platform is excluded.

III. THE PROPOSED SCHEME

Our Scheme design evolves from the token-based access control protocol proposed by Dmitrienko et al. [1]. We revise the original protocol in order to apply it to the specific context of smart home. Such modifications include the introduction of cloud, more specific policy involvement and the replacement of basic communication mode. The cloud-based smart home authentication Scheme consists of eight protocols: platform initialization, device synchronization, user registration, token issuing, owner authentication, token delegation, user(delegated) authentication and token revocation.

A. Objectives

- **Access control**. Only the device owner O or the del.user U who got the valid access token. $T_u$delegated by O have the right to access and send command to the related device.
- **Remote delegation**. Device owner O can remotely grant any person, who have signed in the related cloud, limited right to access his smart home device. “Limited” means that the delegated user can not share his token with others and the device owner O can specific the expire time of the token he authorized. In addition, the life cycle of the delegated token will be terminated when the original token it derived from is updated or abandoned by the device owner.

- **Remote revocation**. Device owner O can authorize cloud to revoke tokens of himself or the delegated tokens he issued before.
- **Cloud-based log query**. Device owner can log in his cloud account to query the information of all devices he can access and manipulate and the history of token delegation.

B. Protocol Specification

Our scheme is composed of the following protocols.

I.Platform Initialization. The device owner O downloads and installs the client application of smart home authentication system which normally operates in the "normal world" $\mathcal{PN}$. Another module in charge of confidential computing and secure storage are installed in the "secure world" in$\mathcal{PS}$. A certificate $cert_p$ and related public key/secure key (pk_p, sk_p) of $\mathcal{PS}$ ought to be secure stored. Subsequent operations such as encryption, decryption and signature must be restricted in $\mathcal{PS}$ as well.

II. Device Synchronization. Several necessary data should be pre-stored in smart home device D before it obtains the permission to enter market: the URL *url* and the certificate $cert_c$ of the cloud-service it will communicate with, an authentication secret K^d_{Auth} ,an encryption key K^d_{Enc} and an unique identification of the device ID_d. When the device is first started up by the device owner O, device D automatically extracts the public key pk_c from $cert_c$ and generates a random number N^d_{syn}, then calculates the ciphertext c^d_{syn} and transfer it to the cloud C.

$$\sigma^d_{syn} \leftarrow MAC(K^d_{Auth}, K^d_{Enc}, N^d_{syn}, ID_d)$$
$$c^d_{syn} \leftarrow Enc(pk_c; K^d_{Auth}, K^d_{Enc}, N^d_{syn}, ID_d, \sigma^d_{syn})$$

After cloud C receives c^d_{syn}, C decryptc^d_{syn} using secret key sk_c. Then it stores K^d_{Auth}, K^d_{Enc} and ID_d on C,then generates a indication$state^c_{syn}$ to notify device D of the synchronization state (success or error occurs). cloud C then calculates a message authentication code σ^c_{syn} and signs it withsk_c. Later then, σ^c_{syn} along with its signature and$state^c_{syn}$ are sent back to device D.

$$\sigma^c_{syn} \leftarrow MAC(ID_d, K^d_{Auth}, K^d_{Enc}, N^d_{syn}, state^c_{syn})$$
$$s^c_{syn} \leftarrow Sig(sk_c; \sigma^c_{syn}, state^c_{syn})$$

Device D receivesσ^c_{syn} and its signature s^c_{syn}, verify s^c_{syn} and re-calculates σ^c_{syn} to make sure all the parameters have not been tampered during the whole negotiation process. If the $state^c_{syn}$ indicates success, device D will keep long-live connection with cloud C for further communication.

III. User Registration. In this phase, device owner O signs himself in cloud C and negotiates the exclusive communication secret key with cloud C. Device owner O inputs his ID ID_o and password pwd_o into his smartphone, then a random number N^o_{reg} is generated in $\mathcal{PS}$. All data along with the certificate of $\mathcal{PS}$ $cert_p$ then are sent to cloud C. Cloud C validates $cert_p$ and (ID_o, pwd_o). Afterwards, an authentication secret K^c_{Auth}, an encryption keyK^c_{Enc} and random number N^c_{reg} are generated by cloud C. Then cloud C generates ciphertext c^c_{reg}. The detail steps to product c^c_{reg} are as follows. (ID_c is the identification number of cloud C)

$$\mathrm{K} \leftarrow \mathrm{MAC}(N^o_{reg}, N^c_{reg}, pwd_o)$$
$$\sigma^c_{reg} \leftarrow MAC(K, ID_o, ID_c, K^c_{Auth}, K^c_{Enc})$$
$$\text{Extract } pk_p \text{ from } cert_p$$
$$c^c_{reg} \leftarrow Enc(pk_p; K^c_{Auth}, K^c_{Enc}, N^c_{reg}, \sigma^c_{reg})$$

After c^c_{reg} is generated and sent back to device owner O's smartphone, In $\mathcal{PS}$, c^c_{reg} is decrypted by sk_p, then K and σ^c_{reg} are re-calculated in order to ensure all the parameters had not been tampered or replayed by malicious adversaries. The total process will abort if the above check fails. After all checks are passed, K^c_{Auth} and K^c_{Enc} are stored in *PS*, and then message authentication code c^o_{reg} are generated using parameters N^c_{reg}, ID_c and ID_o. c^o_{reg} is sent back to cloud C. On receipt of c^o_{reg}, cloud C re-calculated its value to make sure related communication secret key has been well settled. Eventually K^c_{Auth}, K^c_{Enc} are stored with ID_o in C for further usage. So far, the account of device owner O has successfully created on Cloud C. From now on, any operations related with O will be recorded in his log file on C for user's future inspection.

IV. Token Issuing. Device Owner O initiates this section by input his IDID_o and the ID $ID_d$of the smart home device he requires to access and control. Next, an random number N^p_{iss} is generated in $\mathcal{PS}$.N^p_{iss} along with ID_o and ID_d are transferred to cloud C. In cloud C, a list *InUseList* is maintained to record which device's access token has been issued so that cloud C can assure that the access and manipulation right cannot be grant to multiple owners. After cloud C checks ID_d sent by O's smartphone, it generates two authentication secrets $K^{o,d}_{Auth}$, K^{token}_{Auth}, a delegation secret K^o_{Del} and an random number *sn*. K^o_{Del} will be used later to delegate access token. Moreover, $date^o_{exp}$ are set to indicate the expire time of owner O's token. The owner O's access token is generated as follows.

$$m_c \coloneqq (K^o_{Del}, ID_o, K^{o,d}_{Auth}, \mathrm{sn}, date^o_{exp})$$
$$\sigma_c \leftarrow MAC(K^d_{Auth}, m_c, ID_d, K^{token}_{Auth})$$
$$T_o \leftarrow Enc(K^d_{Enc}; m_c, \sigma_c)$$

Further, two extra parameters are generated for secure consideration.

$$\sigma^c_{iss} \leftarrow \mathrm{MAC}(K^c_{Auth}, K^{o,d}_{Auth}, T_o, K^o_{Del}, ID_d, N^p_{iss})$$
$$c^c_{iss} \leftarrow Enc(K^c_{Enc}; T_o, m_c, \sigma^c_{iss})$$

Cloud C adds ID_d into *InUseList* and send c^c_{iss} to device owner O's smartphone. c^c_{iss} is decrypted using K^c_{Enc} stored in $\mathcal{PS}$. If σ^c_{iss} is well verified. Device owner stores$K^{o,d}_{Auth}$, K^o_{Del} and $T_o$in $\mathcal{PS}$.

In the next step, cloud C sends K_{Auth}^{token} to smart home device D for following token authentication through the established long-live connection: Cloud C asks device D to send an random number *N* and its ID ID_d. c_{iss}^{token} is replied.

$$\sigma_{iss}^{token} \leftarrow \text{MAC}(K_{Auth}^{d}, ID_d, ID_o, N, K_{Auth}^{token})$$
$$\sigma_{iss}^{token} \leftarrow \text{Enc}(K_{Enc}^{d}, K_{Auth}^{token}, ID_o, \sigma_{iss}^{token})$$

As smart home device D receives c_{iss}^{token}, D decrypted and verifies it using K_{Auth}^{d}. Eventually, device D stores K_{Auth}^{token} and ID_o. These two parameters will not be changed unless the device owner O requests cloud C to update(re-implement the token issuing process) a new token, then the K_{Auth}^{token} will be replaced by a new value and the former access token will be declared invalid. Until now, Device D has already been bound with device owner O, since the ID of device D has been added into *InUseList*, any token issuing request of device D will be forbidden by Cloud C.

V. Owner Authentication. The identification ID_o of device owner O and the identification $ID_d$of the device O are required to initiate this protocol. Cloud C maintains two revocation List: $RevoList_{id}$ and $RevoList_{dv}$. The $RevoList_{id}$ includes all the owners who have been forbidden to access any smart home device, and the $RevoList_{dv}$ indicates all the (ID_o, ID_d) pairs which are inhibited to connect for now. Cloud C assures that the related parameters from O have not been added into the lists mentioned aboveand delivers O's request to specified smart home device D. D generates an random number $N_{oaut\,h}^{d}$ and replies it to owner's smartphones. In smartphone's $\mathcal{PS}$, random number $N_{oaut\,h}^{o}$ and an indicator Num_{op} that represents the requested operation O are created, and latter parameters are constructed as follows.

$$\sigma_{oaut\,h}^{o} \leftarrow \text{MAC}(K_{Auth}^{o,d}, ID_d, ID_o, N_{oaut\,h}^{d}, N_{oaut\,h}^{o}, Num_{op}, T_o)$$
$$c_{oaut\,h}^{o} \leftarrow \text{Enc}(K_{Enc}^{c}; N_{oaut\,h}^{c}, \sigma_{oaut\,h}^{o}, N_{oaut\,h}^{o}, Num_{op}, T_o)$$

On receipt of $c_{oaut\,h}^{o}$, cloud C decrypts it and transfers the decryption result to smart home device D. Device D then authenticates token T_o and other parameters.

$$(K_{Del}^{o}, ID_o, K_{Auth}^{o,d}, K_{Del}^{o}, \text{sn}, date_{exp}, \sigma_c) \leftarrow \text{Dec}(K_{Enc}^{d}; T_o)$$
$$\sigma_c \stackrel{?}{\Rightarrow} MAC(K_{Auth}^{d}, K_{Del}^{o}, ID_o, K_{Auth}^{o,d}, sn, date_{exp}, ID_d, K_{Auth}^{token})$$
$$\sigma_{oaut\,h}^{o} \stackrel{?}{\Rightarrow} MAC(K_{Auth}^{o,d}, ID_d, ID_o, N_{oaut\,h}^{d}, N_{oaut\,h}^{o}, Num_{op}, T_o)$$

Reject if any of σ_c, $\sigma_{oaut\,h}^{o}$, $date_{exp}$ checks fails, else accept owner O's request and execute the operation Num_{op}. $state_{oaut\,h}$ represents the result of the operation Num_{op}. Smart home device D computes message authentication code$\sigma_{oaut\,h}^{d}$ using $N_{oaut\,h}^{o}$, Num_{op}, ID_d and $state_{oaut\,h}$, further, D sends $(\sigma_{oaut\,h}^{d}, state_{oaut\,h})$ back to owner O through Cloud C.

VI. Token Delegation. Firstly, del.user U ought to register himself to cloud C as protocol III described. When he wants to apply the access right to device D, his identification ID_u, the device identification ID_d, a generated random number N_{del}^{u} and pwd_{del} should be sent to cloud C. pwd_{del} is a one-time secret shared by device owner O and del.user U and ought to be negotiated in advance through confidential out-of-band-channel. Next, cloud C pushes the request and parameters to target device owner O. In the $\mathcal{PS}$ of O's smartphone, two random numbers pn, N_{del}^{o} are generated and an authentication secret $K_{Auth}^{u.d}$ as well. In addition, the expire time $date_{exp}^{u}$ is designated by device owner O. The construction of delegated token T_u and related data are described as follows.

$$m_u \coloneqq (\text{pn}, ID_u, K_{Auth}^{u,d}, date_{exp}^{u})$$
$$\sigma_u \leftarrow MAC(K_{Auth}^{o,d}, m_u)$$
$$T_u \leftarrow Enc(K_{del}^{o}; m_u, \sigma_u)$$
$$\sigma_{del}^{o} \leftarrow \text{MAC}(pwd_{del}, T_u, T_o, ID_d, K_{Auth}^{u,d}, N_{del}^{u}, N_{del}^{o})$$

With the help of cloud C, T_u, T_o, ID_d, $K_{Auth}^{u.d}$, N_{del}^{u}, σ_{del}^{o} are transferred back to del.user U. After data verification in *PS* of U's smartphone, $K_{Auth}^{u.d}T_u$, T_o, $ID_d$are stored. We should notice that all the protocol data transferred between terminals and cloud C in this section must be encrypted by respective secret pair (K_{Auth}^{c}, K_{Enc}^{c}). For example, the data send from the smartphone of U to that of O should first be encrypted by U's secret pair, when the ciphertext arrives cloud C, C decrypts it and then re-encrypts the result using the secret pair belongs to O. Finally, the encrypted data is sent to O and decrypted by the secret pair stored during III.

VII. User(delegated) Authentication. After protocolVI, del.user U gains the limited access right to device D. Whenever U wants to access D, he should send ID_u and both T_u, $T_o$to D. Cloud C acts as an agent between them and promises all checks similar with protocol V passed. the device D receives the message and decrypts T_o to check the its validity and obtain K_{Del}^{o}, which is afterwards utilized to decrypt $K_{Auth}^{u,d}$ from T_u. The rest of the user(delegated) authentication is the same as the process described in V.

VIII. token revocation. There are three methods to revoke owner/user's access right to smart home device D.

- Account revocation. Add user's identification ID_u (ID_o) into $RevoList_{id}$, then his access rights to all devices will be revoked. This revocation is reversible.
- Device revocation. Add the ID pair (ID_u/ID_o, ID_d) into $RevoList_{dv}$, then the user's access right to that single device will be revoked. This revocation is reversible.
- Permanent revocation. smart home device owner applies to cloud C for re-issue the access token of device D, then the former access token along with the delegated token derived from it will become invalid. This revocation process cannot be reversed.

IV. SCHEME SECURITY ANALYSIS

In this section, we will analyze the security of the proposed scheme mentioned above with some common attacks.

Access Token Forgery. The access token of smart home device owner O cannot be forged due to the token is encrypted using unique secret key pair $(K_{Auth}^{d}, K_{Enc}^{d})$ which exclusively belongs to a single smart home device D. This key pair is securely transferred to cloud C encrypted by C's public key in advance. The security of delegated access token depends on the original token it derived from, therefore the security of all owners' tokens are promised, we can deduce that the delegated access token cannot be forged either.

Revocation Determinacy. The service offered by smart home device D cannot be obtained with a revoked or an outdated access token. For temporary revocationVIII(a), VIII(b)), the identifications of user and device are confidential stored in cloud C's revocation lists. Since we assume that C is trusted, revoked device users cannot pass the verification from cloud C. For permanent VIII(c)), the authentication secret K_{Auth}^{token} is replaced by a new value so that the former access token has no opportunity to pass or bypass the check in smart home device D. We should notice that the revocation of device owner's access token will lead to the invalidation of all delegated tokens derived from it.

Replay and Man-in-the-middle Attack. Our scheme can withstand replay and man-in-the-middle attack. Every protocol message communicated between terminals is encrypted by either terminal's public key or secret keys negotiated before. Thus, malicious attackers are impossible to decrypt the message he captured without context information. Moreover, random numbers are involved in each important protocol data transmission which can efficaciously defend replay attacks.

Malware Attack . Malware attack can be performed by installing, modifying arbitrary code on smartphones. However, the "secure world" of ARM TrustZone is isolated from the normal vulnerable execution environment, so the sensitive data such as access token and related encrypt/decrypt computing in “secure world” are immune to malware attack.

Spoofing Attack . Smart home device first establishes connection with cloud using the pre-stored URL and the legal certificate of this cloud, the following communication between device and cloud as well as the communication between cloud and device user are verified by authentication codes shared only within the related two parties. Thus, in every single phase of our scheme, both communication sides could ensure that the object they are talking to is the real target instead of a bogus entity.

V. CONCLUSION

In this paper, we present an efficient authentication scheme for cloud-based smart home system for ubiquitous smart living scenario. Our authentication protocol design is on the basis of electronic access token. Once the access token of a smart home device is issued to a certain individual, he can use this token multiple time to access and manipulate the target device. The authentication process therefore is largely simplified compared with the conventional username-password method. In addition, the scheme we proposed introduced some more flexible functions. A smart home device can bound to only one device owner but multiple device user can be delegated by the owner with limited access right to this device. The device owner limits the usage of delegated tokens by setting token expire time and undergoing revocation process in certain situations. Moreover, security is strictly guaranteed by the implementation of ARM TrustZone, certificate architecture, electronic access token and pre-negotiated secret key pairs.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under Grant No.61202476.

REFERENCES

[1] Dmitrienko, A., Sadeghi, A.R., Tamrakar, S.,Wachsmann, C., “SmartToken: Delegable Access Control with NFC-enabled Smartphones,” In: 5th International Conference on Trust & Trustworthy Computing(2012).

[2] Vaidya, B., Park, J.H, Yeo, S.S., Rodrigues, J.J., “Robust one-time password authentication scheme using smart card for home,” In: Computer Communications(2011).

[3] Ibriq, J., Mahgoub, I, “A Hierarchical Key Establishment Scheme for Wireless Sensor Networks.” 21st International Conference on Advanced Networking and Applications(2007).

[4] Bickford, J., O'Hare, R., Baliga, A., Ganapathy, V., Iftode, L., “Rootkits on Smart Phones: Attacks, Implications and Opportunities,” In: HotMobile'10 Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications(2010).

[5] Alves, T., Felton, D., “TrustZone: Integrated hardware and software security,” In: Information Quaterly,3(4)(2004).

[6] Zhou, L., Varadharajan, V., Hitchens, M., “Achieving Secure Role-Based Access Control on Encrypted Data in Cloud Storage,” In: IEEE Transactions on Information Forensics and Security(2013).

[7] Tsuji, T., Shimizu, A., “One-time password authentication protocol against theft attacks, ”In: IEICE Transactions on Communications(2004).

[8] Brasser, F.F., Bugiel, S., Filyanov, A., Sadeghi, A., Schulz, S., “Softer Smartcards Usable Cryptographic Tokens with Secure Execution,” In: Dependable Systems and Networks(DSN), pp. 1-12. IEEE(2011).

[9] Filyanov, A., McCune, J.M., Sadeghi, A.R., Winandy,M., “Uni-directional trusted path: Transaction confirmation on just one device, ”In: IEICE Transactions on Communications(2004).

[10] Robles, R.J., Kim, T., “A Review on Security in Smart Home Development,” In: International Journal of Advanced Science and Technology(2010).

[11] Ye, X., Huang, J., “A Framework for Cloud-based Smart Home, ”In: International Conference on Computer Science and Network Technology(2011).