Analysis on Secured Communication using QSNFC Protocol on Internet of Things

[1] Amruta Gadekar
[1] Assistant Professor
[1]Department of Computer Engineering,
[1]D.Y.Patil Engineering College, Pune, India

Abstract : The data used on Internet or online related application should be more secure if that is used for payment or access control. This security is provided by Near Field Communication (NFC) Protocol. The NFC is used for a wide range of security-critical applications such as payment or access control. Such applications require secured data transfer, but NFC protocol does not include transport layer security, that's why provide insufficient security measures. To ease NFC usage in the Internet of Things (IoT) where millions of devices need to be secured, an efficient and sufficiently secured NFC-based protocol needs to be developed. This paper introduces the more secured NFC protocol which is capable of performing more efficient key agreements for recurring connections, and thus, can be used as an efficient alternative to the Transport Layer Security (TLS) protocol.

***IndexTerms* - Internet of Things, Near Field Communication,Transport Layer Security.**

I. Introduction

The Internet of Things (IoT) provides any service, any time and connect the people and the things using any devices. The internet of things is the new revolution of internet which rapidly changes. The devices that we uses in a wide range of areas such as smart homes, transportation, healthcare, or industrial scenarios. The Internet of Things provides interaction among the real/physical and the digital/virtual worlds. The physical entities have digital counterparts and virtual representation and things become context aware and they can sense, communicate, interact, exchange data, information and knowledge.

In this rapid growth of IoT,each time it uses several technologies.This technology developments in Radio Frequency Identification (RFID), smart sensor technology, and communication technologies and protocols as such enabling technologies.With these technology one more new technology introduces is the NFC as a very promising technology for the IoT.

NFC is the Near Field Communication used wireless data transfer between two portable devices. Near Field Communication(NFC) as a very promising technology for the IoT since many smartphones nowadays are equipped with NFC-enabling technology.NFC is basically used in contactless payments.

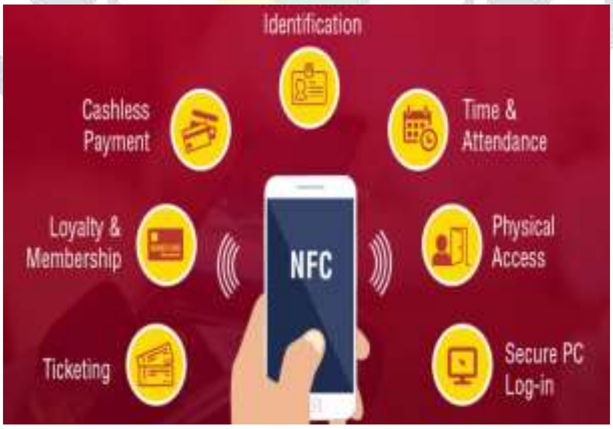


fig.1 : NFC use

In the IOT related communication protocol the security is most important aspect if it is used for payment and access control.But in IOT related protocol that security is oftenly neglected by the protocol.If transferred data is protected by weak security measures or even transferred unprotected, attacks are threatening the confidentiality of critical information then it is measure issue in the IoT.By considering these issues this paper introduces the security NFC protocol which will require in fewer messages to be exchanged during key agreement.

II. RELATED WORK

Al-Fuqaha et al. In [1] this paper authors introduces new technologies are used in IoT such as RFID, smart sensors, communication technologies, and Internet protocols. Together with RFID, the authors also mention Near Field Communication (NFC) as a very promising technology for the IoT since many smartphones nowadays are equipped with NFC-enabling technology. In[2] this paper the authors focuses on security and privacy related issues in IoT. For development of security mechanism and confidentiality authors introduces security triad or CIA(Confidentiality,Integrity,Availability) triad, a distinguished model. Cyberattacks on IoT systems are very critical since they may cause physical damage and even threaten human lives. The goals of IoT are to ensure proper identity authentication mechanisms and provide confidentiality about the data etc. The Security triad or CIA triad, a distinguished model for the development of security mechanisms

In [3] authors gives an introduction to Industrial IoT systems, the related security and privacy challenges related to Cyber-attacks on IoT systems which are very critical since they may cause physical damage and even threaten human lives. In[4] the paper introduces the information and the use of NFC. Haselsteiner and Breitfuß demonstrate that eavesdropping data is possible up to 10m. In [5] Pl´osz et al compare the provided security of various wireless communication technologies with NFC regarding the data transfer security. In [6] Chen et al. list a large number of attacks that are feasible for attackers to perform due to weak or insufficient security in the NFC protocol.

III. System model and quick & secured NFC

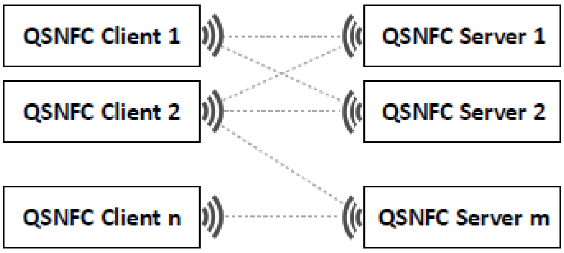


fig.2 : system model

In fig.2 As shown in this model, the protocol is supposed to support an arbitrary number of QSNFC clients as well as an arbitrary number of QSNFC servers.

A. QSNFC client: The QSNFC client is the entity that tries to establish a secured communication channel with the QSNFC server. Since this entity is initiating the NFC communication, it can be seen as the active component in NFC terms.

B. QSNFC Server: The QSNFC server is contacted by the QSNFC client in order to establish a secured communication channel. In NFC terms, the QSNFC server would be denoted as the passive component.

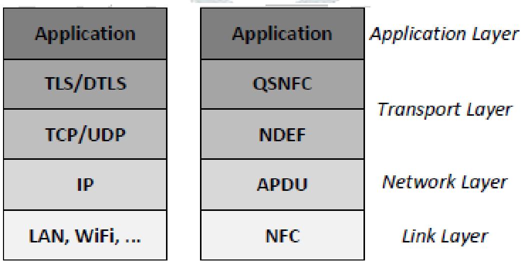
C. Communication channel: The communication channel is an NFC channel with the potential presence of an undetected adversary.

D. Adversary: The adversary present is assumed to be capable of eavesdropping and modifying ongoing NFC data.

IV. OUICK AND SECURED NFC PROTOCOL

4.1 Layer Model

To know the classification of QSNFC is important. The QSNFC classification is similar to the TCP/IP protocol architecture layer. The similarities to TLS and DTLS ,the QSNFC protocol resides directly underneath the actual application and provides capabilities for secured data transfer to the upper layer.



Application	Application	Application Layer
TLS/DTLS	QSNFC	Transport Layer
TCP/UDP	NDEF	
IP	APDU	Network Layer
LAN, WiFi, ...	NFC	Link Layer

fig.3: protocol stacks for TLS/DTLS and QSNFC

As a transport protocol, the NFC Data Exchange Format (NDEF) that is based on application protocol data unit (APDU) packets is used. NDEF itself provides limited security measures, such as signature records. However, these security measures are shown to be vulnerable to certain attacks. Therefore, we use NDEF as a transport protocol for QSNFC only, without relying on any of the available security features of NDEF. Both TLS/DTLS and QSNFC reside underneath the application layer and provide their functionality to higher layers, while relying on lower-layer protocols to perform data transfer.

4.2 STRUCTURE OF QSNFC

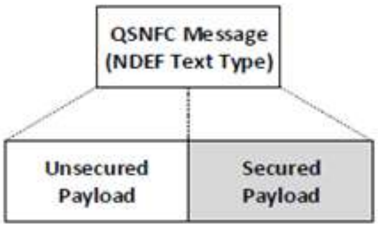


fig.4:Basic structure of QSNFC

The basic packet structure of each QSNFC message is shown in Fig. As can be seen there, any QSNFC message comprises unsecured as well as secured payload inside a text record type NDEF message. A text record type NDEF message comprises unsecured and secured payload.

4.3 CONNECTION ESTABLISHMENT

To data transfers using QSNFC based on secured data channels, key agreement needs to be performed. For recurring connections, the client needs to cache information about the server if a successful initial handshake is performed. Fig. shows the the handshake process in comparison to TLS.

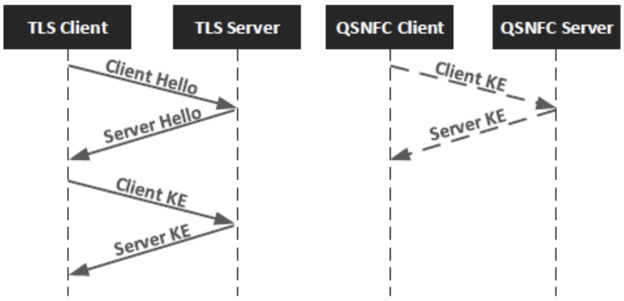


fig.5: Round trips required for TLS and QNFC

Initial handshake: To initiate this handshake, the client sends a so-called inchoate client hello (CH) message to the server, which recognizes the inchoate information and replies with a reject (RJ) message.
(i) The server's long-term DH publicvalue. This public key is used for the generation of subsequent keys and thus,needs to be cached by the client.
(ii) A certificate chain that authenticates the server and that needs to be verified during the initial handshake.
(iii) A signature of the long-term DH public value that is signed using the private key from the provided certificate chain's leaf certificate.
(iv) A source address token that contains the server's unique ID and a nonce from the server. This information is protected using AE. The client needs to send this token back to the server in subsequent handshakes to demonstrate ownership of the server's identity.

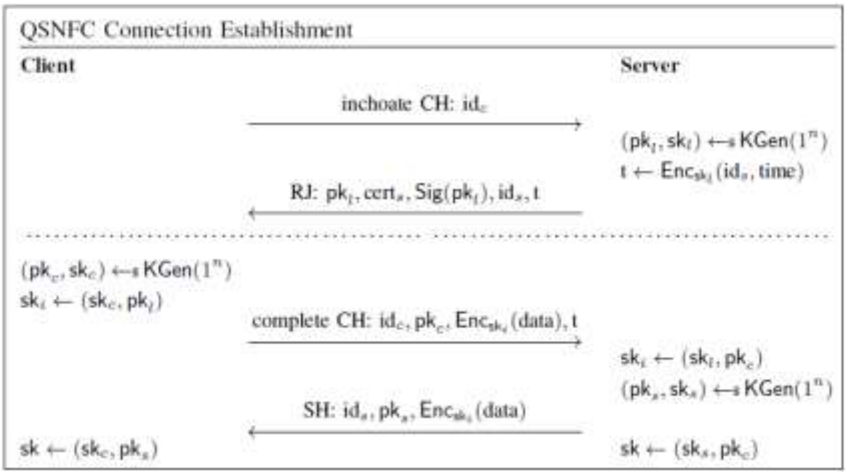

QSNFC Connection Establishment
Client
Server
inchoate CH: id_c
$(\mathsf{pk}_\ell, \mathsf{sk}_\ell) \leftarrow\!\!\$\ \mathsf{KGen}(1^n)$
$\mathsf{t} \leftarrow \mathsf{Enc}_{\mathsf{sk}_l}(\mathsf{id}_s, \mathsf{time})$
RJ: $\mathsf{pk}_\ell, \mathsf{cert}_s, \mathsf{Sig}(\mathsf{pk}_\ell), \mathsf{id}_s, \mathsf{t}$
$(\mathsf{pk}_c, \mathsf{sk}_c) \leftarrow\!\!\$\ \mathsf{KGen}(1^n)$
$\mathsf{sk}_i \leftarrow (\mathsf{sk}_c, \mathsf{pk}_\ell)$
complete CH: $\mathsf{id}_c, \mathsf{pk}_c, \mathsf{Enc}_{\mathsf{sk}_i}(\mathsf{data}), \mathsf{t}$
$\mathsf{sk}_i \leftarrow (\mathsf{sk}_\ell, \mathsf{pk}_c)$
$(\mathsf{pk}_s, \mathsf{sk}_s) \leftarrow\!\!\$\ \mathsf{KGen}(1^n)$
SH: $\mathsf{id}_s, \mathsf{pk}_s, \mathsf{Enc}_{\mathsf{sk}_i}(\mathsf{data})$
$\mathsf{sk} \leftarrow (\mathsf{sk}_c, \mathsf{pk}_s)$
$\mathsf{sk} \leftarrow (\mathsf{sk}_s, \mathsf{pk}_c)$

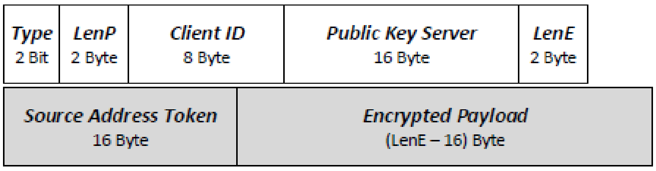
fig.6 : QSNFC connection establishment

Subsequent handshake : The client already is in possession of the server's long-term DH public value, it can calculate a shared key using its own ephemeral DH public value. The client can then send a complete CH, without first sending a inchoate CH message as is done in the initial handshake. Thus, the first RTT from the initial handshake is not required and encrypted data can be sent to a known server instantly.

If the handshake is successful, a server hello (SH) message is sent by the server as response to the complete CH. The SH is encrypted using a key generated from the server's long-term public DH public value and the client's ephemeral DH public value. The SH message also contains the server's ephemeral DH public value. The complete connection establishment is shown in Fig. 6. After both involved entities are in possession of each others' ephemeral DH public value, a forward-secure key can be calculated for the connection. Thus, after the SH message is sent and received, both communicating entities switch to encrypting packets with the forward-secure keys.

V. PACKET STRUCTURE

5.1 CH messages: It can be either an inchoate CH message or a complete CH message. If a client initiates the initial handshake by sending an inchoate CH message, the Client ID is set accordingly with each other attribute being empty. Client ID is a unique identifying the client.

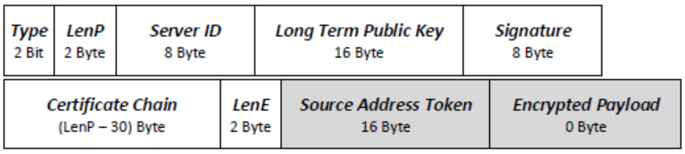


Type 2 Bit	LenP 2 Byte	Client ID 8 Byte	Public Key Server 16 Byte	LenE 2 Byte

Source Address Token 16 Byte	Encrypted Payload (LenE – 16) Byte

fig.7a: CH packet structure

5.2 RJ messages: RJ messages are sent as a response to inchoate CH messages. The message contains information from the server that is required to perform connection establishment and key agreement. Server ID: A unique ID identifying the server. The long-term DH public value that is used for key agreement and to calculate initial keys. A signature of the Long Term Public Key that is used by the client to validate the integrity of that key. The certificate chain that is used by the client to authenticate the server. The certificate chain that is used by the client to authenticate the server. In RJ messages, only the Source address token is contained in the encrypted payload.

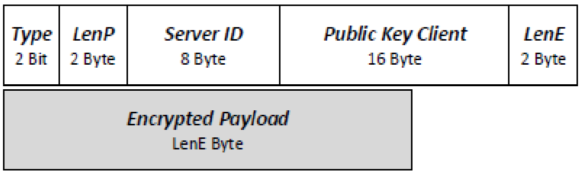


Type 2 Bit	LenP 2 Byte	Server ID 8 Byte	Long Term Public Key 16 Byte	Signature 8 Byte

Certificate Chain (LenP – 30) Byte	LenE 2 Byte	Source Address Token 16 Byte	Encrypted Payload 0 Byte

fig.7b: RJ packet structure

5.3 SH messages: SH messages are sent by the server in response to a successful connection establishment. The Server is unique ID is used by the client to match cached information to the correct server.

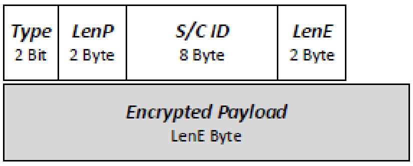


Type 2 Bit	LenP 2 Byte	Server ID 8 Byte	Public Key Client 16 Byte	LenE 2 Byte

Encrypted Payload LenE Byte

fig.7c: SH packet structure

5.4 SD messages:SD messages are exchanged between server and client after successful handshakes. S/C ID: The respective ID of either server or client is included to identify the sender of an SD message. Payload of arbitrary length that is protected by AE using the previously established ephemeral forward-secure keys.



Type 2 Bit	LenP 2 Byte	S/C ID 8 Byte	LenE 2 Byte

Encrypted Payload LenE Byte

fig.7d: SD packet structure

IV. COMPARATIVE ANALYSIS

The following tables shows the comparison with the online application and comparison among the protocols.

Table 6.1: Comparative Analysis

Parameters	Confidentiality	Integrity	Authenticity	Standardized	Efficient
Secret Sharing	✓	✗	✗	✗	✗
Secure UHF-RFID Tag	✓	✗	✓	✗	✗
Mobile payment	✓	✓	✓	✗	✗
Healthcare	✓	✓	✓	✗	✗
Car immobilizer	✓	✓	✓	✗	✗
TLS over NFC	✓	✓	✓	✓	✗
QSNFC	✓	✓	✓	✓	✓

IIV. CONCLUSION

For the secure communication NFC based data transfer is used. QSNFC protocol is designed for quick and secured data transfer for payment or access control. The protocol fulfils the 0-RTT requirement to increase the performance of recurring connections between devices. Data confidentiality, integrity, and authenticity for transferred data is provided.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] M. U. Farooq, M. Waseem, A. Khairi, and S. Mazhar, “A Critical Analysis on the Security Concerns of Internet of Things (IoT),” International Journal of Computer Applications, vol. 111, no. 7, 2015.

[3] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, “Security and Privacy Challenges in Industrial Internet of Things,” in Design Automation Conference (DAC), 52nd ACM/EDAC/IEEE. IEEE, 2015, pp. 1–6.

[4] L. M. R. Tarouco, L. M. Bertholdo, L. Z. Granville, L. M. R. Arbiza, F. Carbone, M. Marotta, and J. J. C. de Santanna, “Internet of Things in Healthcare: Interoperability and Security Issues,” in Communications (ICC), IEEE International Conference on. IEEE, 2012, pp. 6121–6125.

[5] L. Finˇzgar and M. Trebar, “Use of NFC and QR code Identification in an Electronic Ticket System for Public Transport,” in Software, Telecommunications and Computer Networks (SoftCOM), 19th International Conference on. IEEE, 2011, pp. 1–6.

[6] E. Haselsteiner and K. Breitfuß, “Security in Near Field Communication (NFC),” in Workshop on RFID security, 2006, pp. 12–14.

[7] C. H. Chen, I. C. Lin, and C. C. Yang, “NFC Attacks Analysis and Survey,” in Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2014 Eighth International Conference on. IEEE, 2014, pp. 458–462.

[8] N. A. Chattha, “NFC - Vulnerabilities and Defense,” in Information Assurance and Cyber Security (CIACS), 2014 Conference on. IEEE, 2014, pp. 35–38.