Security
CoAP is built on top of UDP not TCP, so SSL/TLS are not available to provide security. DTLS (Datagram Transport Layer Security) provides the same assurances or security as TLS but for transfers of data over UDP. Most commonly, DTLS capable CoAP devices are able to support RSA and AES or ECC and AES.
Security Protocol & Application for CoAP
CoAP is most common and now becoming the standard protocol for IoT applications. Security is an important aspect to protect the communication between IoT end-node devices. In the following part, a security protocol DTLS is introduced to secure the CoAP and other UDP based protocol. Also, one of CoAP application, Smart Homes, describes to the case study.

Why use DTLS for CoAP Security
There are three main elements when considering security, named as integrity, authentication and confidentiality. DTLS can achieve all of them. IETF modifies TLS to develop another security protocol DTLS. DTLS employ on TCP/UDP, which is too complex. DTLS solves two problems: reordering and packet lost. It adds three implementations: 1) packet retransmission. 2) assigning sequence number within the handshake. 3) replay detection.

DTLS stays in application layer (Fig. 4) and protect end-to-end communication just like network layer security protocols. It’s not easy for attackers to access any data from end-to-end communication where data passes through a compromised node. DTLS also avoids cryptographic overhead problems with some exception that usually occurs in lower layer security protocols.

![Diagram of DTLS protocol stack](image)

**Fig 4: DTLS in protocol stack**

Structure of DTLS
There are two layers in DTLS. The bottom one contains Record protocol. The upper one includes three protocols which are Alert, Handshake and application data, in some condition Change Cipher Spec protocol may replace one of them. The Change Cipher Spec message is used to notify Record protocol to protect subsequent records with just-negotiate cipher suite and keys.
Record protocol protects application data by using keys generated during Handshake. For outgoing messages, protocol divides, compress, encrypt and apply Message Authentication Code (MAC) to the messages. For incoming messages, protocol reassemble, decompress, decrypt and verify them based on various algorithm used on source and destination. Record header is made of two parts, one is content type and another is fragment field. Content type decides what data or content is going to be contained in fragment field. It could be a handshake protocol, alert protocol or an application data. In the Comparison with DTLS Record, Handshake protocol is rather a complex one which involves a lot of exchange steps. Individual messages are grouped into message flights. Fig. 5 shows the process of Handshake.

![Fig. 5: Process of Handshake](image)

The Architecture shown below (Fig. 6) is for uni-cast communication (client interacts with one or more servers). The client needs to know which certificate or public key it has to use with which server.
Fig. 6: Uni-cast communication model
Research Area

This research will be around most common IOT communication protocols – COAP.

i. Security design and structure analysis on network, transport and application layers.

ii. Encryption mechanism at application and network layers.

iii. Communication security.

iv. Cryptographic Algorithms evaluation and appropriate selection.

v. Key-cert security process analysis.

vi. Evaluation of existing and new implementation.

vii. Analyze a security and interoperability level which can provide a measurement to security level.

viii. Analysis of various layers of communication in the respect to security.