Appendix A

Model Code of the Proposed MQTT Protocol

1. There are few parts if the ProVerif Compiler

a. **Declarations**: It is the parameters where all the channels and associated variables are declared.

b**. Functions** : All built in functions are defined here.

c. **Process**: These are the functions which are written in order to carry out the testing. There are three main functions – Broker Process, Client Process and authentication server process.

d. **Main Process**: This is the main function from where all the local functions are being called to get the results.

e. **Secrecy query**: Main Queries are outlined in this part.

2. **Declarations (Public channels and data)**

free Client_Broker_Public_Ch:channel.

free Client_AuthenticationServer_Public_Ch: channel.

free Broker_AuthenticationServer_Public_Ch: channel.

free user_id: bitstring.

free user_type: bitstring.

free broker_id: bitstring.

```
Free broker_type: bitstring.
```

3. **Functions description**

```
fun Enc(bitstring, bitstring): bitstring. (*constructor*)

reduc forall x:bitstring, y:bitstring; dec(Enc(x,y),y) = x. (*destructor*)

fun Hash(bitstring,bitstring): bitstring.

fun Concat(bitstring,bitstring): bitstring.

fun Deconcat(bitstring, bitstring):bitstring.
```

4. **Private data which secrecy is verified**

```
free session_key: bitstring [private].

free user_secret_credential: bitstring [private].

free broker_secret_credential: bitstring [private].
```

5. **Secrecy query**

```
query attacker(session_key).

query attacker(user_secret_credential).

query attacker(broker_secret_credential).
```

6. Client Process

```
let Client =

   new R_u: bitstring;

   let X_u= Enc(R_u, user_secret_credential) in
```

```
    out (Client_Broker_Public_Ch, (user_id, user_type, X_u));
in (Client_Broker_Public_Ch, (b_id_user: bitstring, b_type_user: bitstring, X_b_user: bitstring,I_u_user: bitstring));
 let I_u= Hash(X_b_user, user_secret_credential) in
out(Client_AuthenticationServer_Public_Ch,(b_id_user,b_type_user,X_b_user,user_id, user_type));
 in(Client_AuthenticationServer_Public_Ch,(b_id_ua:bitstring,b_type_ua:bitstring, broker_type_a:bitstring,A_R_ba:bitstring,I_ab_a:bitstring,I_a_ua:bitstring));
    let R"_b_s= dec(A_R_ba, user_secret_credential) in
    let I'_au= Hash(R"_b_s, user_secret_credential) in
    let R"_b= Deconcat(R"_b_s, session_key) in
    out(Client_Broker_Public_Ch,(user_id, user_type,R"_b, I_ab_a)); 0.
```

7**. Broker Process**

```
let Broker =
    in (Client_Broker_Public_Ch, (u_id_broker: bitstring, u_type_broker: bitstring, msg: bitstring));
    out(Broker_AuthenticationServer_Public_Ch,(u_id_broker,u_type_broker,msg, broker_id, broker_type));
    in (Broker_AuthenticationServer_Public_Ch, (user_id_a : bitstring, user_type_a: bitstring, A_R_a: bitstring,
    I_u_a: bitstring, I_b_a: bitstring ));
    let R"_u_s= dec(A_R_a, broker_secret_credential) in

    let I"_b= Hash(R"_u_s, broker_secret_credential) in

    let R"_u= Deconcat(R"_u_s,session_key) in

    new R_b: bitstring;

    let X_b= Enc(R_b, broker_secret_credential) in

    out(Client_Broker_Public_Ch,(broker_id,broker_type,X_b,I_u_a));

    in(Client_Broker_Public_Ch,(user_id_b:bitstring, user_type_b:
bitstring,R"_b_a:bitstring, I_ab_user:bitstring ));

    let I"_ab= Hash(R"_b_a, broker_secret_credential) in 0.
```

8. **Authentication Server Process**

```
let Auth =     in (Broker_AuthenticationServer_Public_Ch, (uid_b: bitstring, utype_b:
bitstring, msg_cb: bitstring, bid: bitstring,    btype: bitstring));

    let R_u_a= dec(msg_cb, user_secret_credential) in

    let I_u= Hash(R_u_a, user_secret_credential) in

    let A_R_u = Enc(Concat(R_u_a, session_key), broker_secret_credential) in

    let I_b = Hash(Concat(R_u_a, session_key), broker_secret_credential) in

    out (Broker_AuthenticationServer_Public_Ch, (uid_b, utype_b,A_R_u, I_u, I_b));

    in
(Client_AuthenticationServer_Public_Ch,(b_id_u:bitstring,b_type_u:bitstring,X_b_u:bitst
ring,user_id_u:bitstring,
    user_type_u:bitstring));

    let R"_b= dec(X_b_u, broker_secret_credential) in

    let I_ab= Hash(R"_b, broker_secret_credential) in

    let A_R_b = Enc(Concat(R"_b, session_key), user_secret_credential) in

    let I_a_u = Hash(Concat(R"_b, session_key), user_secret_credential) in

    out(Client_AuthenticationServer_Public_Ch,(b_id_u,b_type_u,
broker_type,A_R_b,I_ab, I_a_u)).
```

9. **Main process**

```
 !Client | !Broker | !Auth
```