profile.c

```
init_interface_struct() {
    Set default dest.
    allocate interface struct
    clear initial state
    create an empty state machine
    eapol_init( interface )
    clear config file data (user)
}
```

os_cardif.c

```
get_frame(interface, frame) {
    Get frame from the network
}
sendframe(interface, frame ) {
    Send frame to the network
}
```

config.c

```
config_setup() {
    Parse Config File
}
config_build() {
    Populate local data structure
    with config data
}
```

key_statemachine.c

```
run_key_statemachine() {
}
```

eapol.c

```
eapol_execute(int_list  ) {
    getframe( int_list  frame
    Parse Frame
    statemachine_run(newframe
                     respframe)
    sendframe(respframe)
}
eapol_init( interface  ) {
    initialize statemachine vars
    eap_init( interface )
}
```

statemachine.c

```
statemachine_run() {
    Process State Machine vars
    transmit necessary packets

txRspId() {
    setup Response header
    eap_request_id()
}
txRspAuth() {
    setup Response header
    eap_request_auth()
}
txStart() {
    Setup Start Frame
}
txLogoff() {
    Setup logoff frame
}
```

xsupp_driver.c

```
int_list =NULL;

global_init() {
    init_interface_struct( int_list )
    config_setup()
    config_build()
}

main() {
    Argument checking
    global_init()
    while (1) {
        eapol_execute( int_list )
    }
}
```

eap.c

```
eap_process_header() {
    Parse EAP Frame
    Update  int_list
}
eap_init( interface   ) {
}
eap_request_id() {
    Setup EAP Header
}
eap_request_auth() {
    Determine which method
    If first message {
        <method>.eap_auth_setup()
    }
    <method>.eap_auth_handler()
}
```

<method_name>.c

```
eap_auth_setup() {
}
eap_auth_handler() {
}
```