WEP Security Application

User guide
Version 0.1
March 2016

Redpine Signals, Inc.

2107 N. First Street, #680

San Jose, CA 95131.

Tel: (408) 748-3385

This project is applicable to all the WiSeConnect variants like WiSeConnect Plus, WiSeMCU and WYZBEE. The term WiSeConnect refers to its appropriate variant.

Application Overview:

Wired Equivalent Privacy (WEP) is a security protocol for wireless networks that encrypts transmitted data . The disadvantage is that without any security, your data can be intercepted without difficulty.

WEP has three settings: Off (no security), 64-bit (weak security), 128-bit (a bit better security). WEP is not difficult to crack, and using it reduces performance slightly.

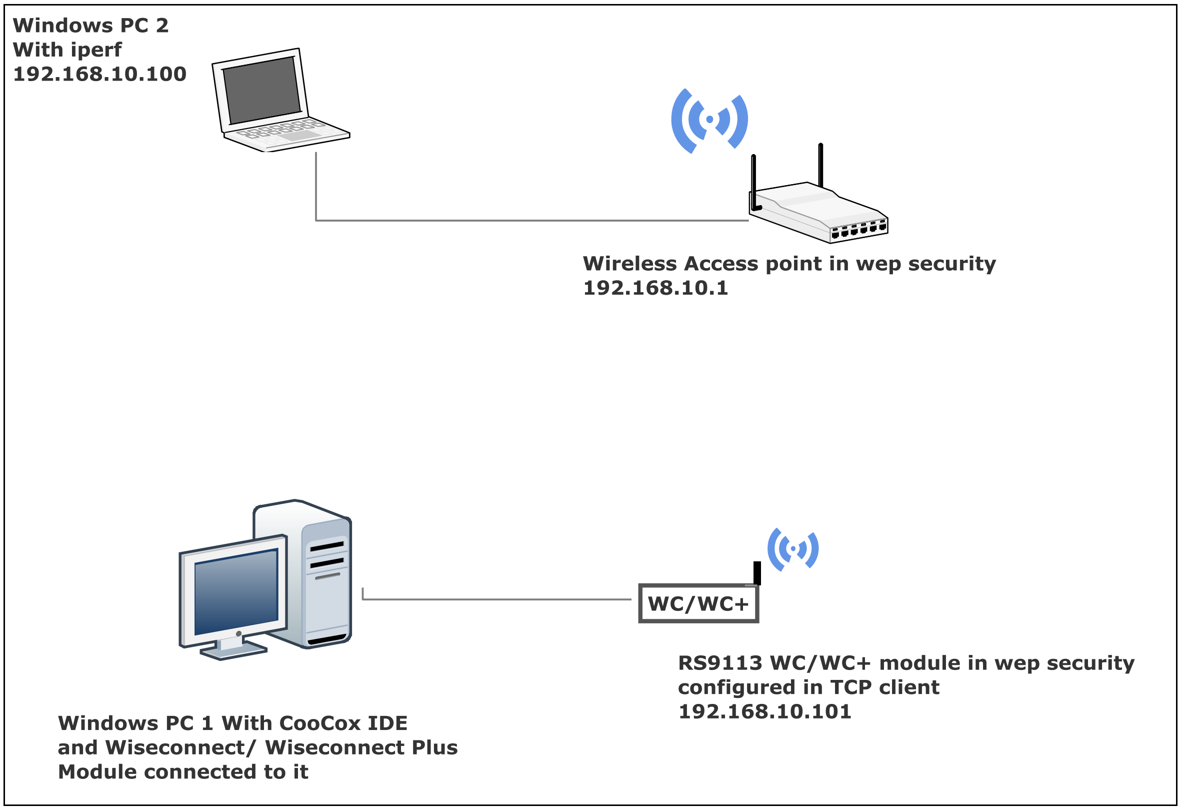
However, WEP was an early attempt to secure wireless networks, and better security is now available such as DES, VPN, and WPA

The WEP security application demonstrates how to connect to a WEP secured Access point and open a standard TCP client socket and sends data on server socket with WiSeConnect device. Following are the procedure steps

- Configure an Access point in wep secured mode.
- Connect the Wise connect device to the wep secured AP and get IP through DHCP
- Server socket at Access point has to be opened by means of application program like iperf.
- TCP client socket would be connected to the server socket that is opened.
- Once the connection is established, Data is sent to the TCP server.

Setup required:

1. Windows PC with Coocox IDE
2. WiSeConnect device
3. Access Point with WEP security
4. Application program like iperf


Windows PC 2
With iperf
192.168.10.100
Wireless Access point in wep security
192.168.10.1
WC/WC+
RS9113 WC/WC+ module in wep security
configured in TCP client
192.168.10.101
Windows PC 1 With CooCox IDE
and Wiseconnect/ Wiseconnect Plus
Module connected to it

Description:

This application is used to demonstrate WEP connectivity and open a TCP client socket in WiSeConnect device.

Once it is configured as TCP client it can establish and maintain a network conversation by means of application program for exchanging of data.

Configuring the Application:

Edit the **`rsi_wep_security.c`** file in the following path .

`sapis/examples/wlan/wep_security/`

1. From given configuration,

   `SSID` refers to the name of the Access point to connect.

   `CHANNEL_NO` refers to particular channel used to scan by the device.If channel is 0 then it will scan all channels.

   `SECURITY_TYPE` refers to type of security WEP (RSI_WEP)

   `WEP_INDEX` refers to the one of the four keys to be used for connection

WEP_KEY0, **WEP_KEY1**, **WEP_KEY2**, **WEP_KEY3** are the keys , they can be 10 bytes or 26 bytes.

```
#define SSID                 "<ap_name>"
#define CHANNEL_NO            <channel_num>
#define SECURITY_TYPE         RSI_WEP
#define WEP_INDEX             "<index>"
#define WEPKEY0               "<wep key 0>"
#define WEPKEY1               "<wep key 1>"
#define WEPKEY2               "<wep key 2>"
#define WEPKEY3               "<wep key 3>"
```

1. Enable/Disable DHCP mode

   1 – Enables DHCP mode (gets the IP from DHCP server)

   0 – Disables DHCP mode

```
#define  DHCP_MODE           <dhcp mode>
```

2. To configure static IP address

   IP address to be configured to the device should be in long format and in little endian byte order.

   Example: To configure "192.168.10.1" as IP address, update the macro **DEVICE_IP** as **0x010AA8C0**.

```
#define  DEVICE_IP              0X010AA8C0
```

   IP address of the gateway should also be in long format and in little endian byte order

   Example: To configure "192.168.10.1" as Gateway, update the macro **GATEWAY** as **0x010AA8C0**

```
#define  GATEWAY                0x010AA8C0
```

   IP address of the network mask should also be in long format and in little endian byte order

   Example: To configure "255.255.255.0" as network mask, update the macro **NETMASK** as **0x00FFFFFF**

```
#define   NETMASK               0x00FFFFFF
```

3. To establish TCP connection and transfer data to the remote socket configure the below macros

   Internal device port number

```
#define DEVICE_PORT             <Internal_port_num>
```

   Port number of the remote server

```
#define SERVER_PORT             <Remote_port_num>
```

IP address of the remote server

```
#define SERVER_IP_ADDRESS       0x640AA8C0
```

Number of packet to send

```
#define NUMBER_OF_PACKETS     <num_of_packets>
```

Application memory length which is required by the driver

```
#define  GLOBAL_BUFF_LEN        8000
```

Update the Wlan configuration file:

sapis/include/rsi_wlan_config.h

CONCURRENT_MODE	DISABLE
RSI_FEATURE_BIT_MAP	FEAT_SECURITY_OPEN
RSI_TCP_IP_BYPASS	DISABLE
RSI_TCP_IP_FEATURE_BIT_MAP	(TCP_IP_FEAT_DHCPV4_CLIENT)
RSI_CUSTOM_FEATURE_BIT_MAP	0
RSI_BAND	RSI_BAND_2P4GHZ

Executing the Application:

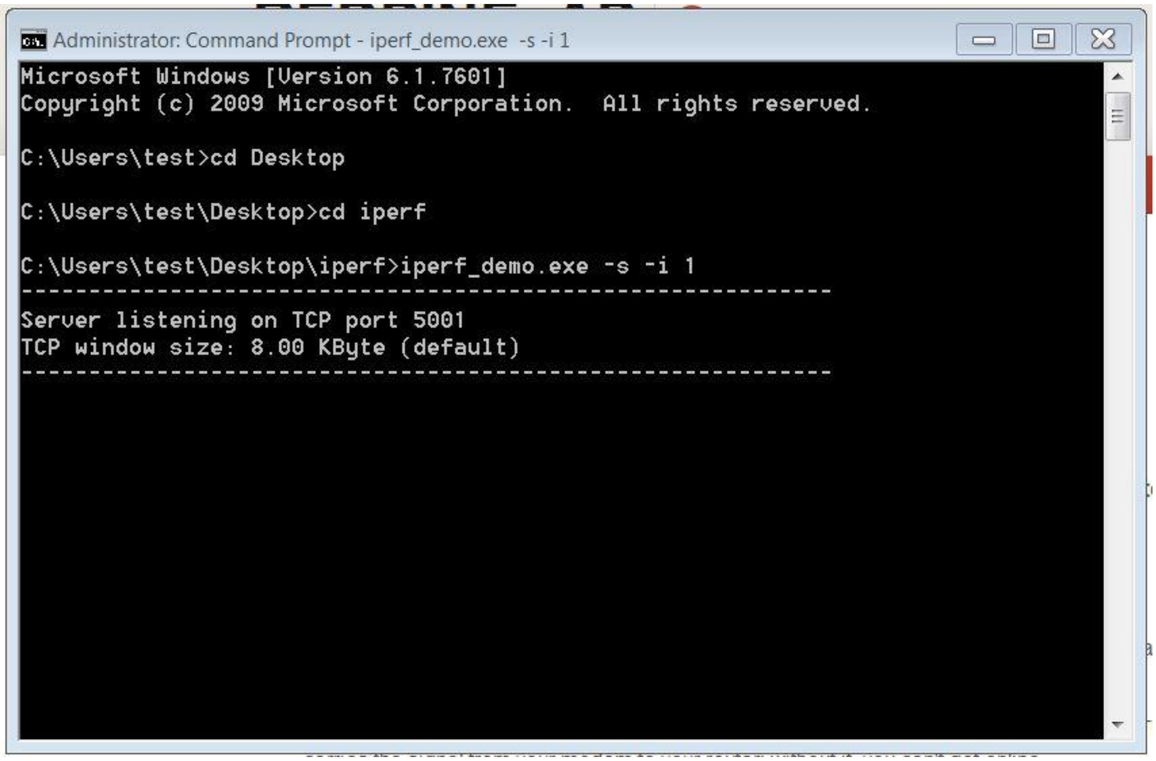
1. Connect WiSeConnect device to the Windows PC running Cocoox IDE.
2. Configure the macros in the files located at

   sapis/examples/wlan/wep_security/rsi_wep_security.c

   sapis/include/rsi_wlan_config.h

3. Open an iperf TCP server listening on port SERVER_PORT on remote machine in the following format.

```
iperf.exe -s -p <SERVER_PORT> -i 1
```



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\test>cd Desktop

C:\Users\test\Desktop>cd iperf

C:\Users\test\Desktop\iperf>iperf_demo.exe -s -i 1
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
```

4. Build and launch the application.

5. After the program gets executed, WiSeConnect Device would be connected to Access point having the configuration same that of in the application and get IP.

6. The Device which is configured as TCP client will connect to iperf server and sends number of packets configured in `NUMBER_OF_PACKETS`