I have neither given or received any unauthorized aid on this assignment.

# Defenses for a Diabetes Therapy System from Security Attacks

This work is aiming to improve the security in data transmission between different medical device components through wireless connection. Because of the wireless connection and easy access to the critical information of the system, the device is vulnerable to malicious attacks. As reported, the only mechanism used in a diabetes therapy system is the cyclic redundancy check, namely, when the current count values is equal to the previous value, the current data package is rejected. It does not encrypt its 36-bit PIN numbers. In this report, the rolling code technique will be adopted to improve the PIN security level. 8 keys are applied in a sequence which is indicated by a sequence counter. The PIN of the original data package is encrypted and then sent to the insulin pump based on the algorithm described in this report. The rolling encoder and decoder is shown as in Figs. 1 and 2. In addition, the report system will be also suggested to connect the insulin pump system to a phone to display the count value and trigger alarm system of the phone when the count value is unreasonable low or high.

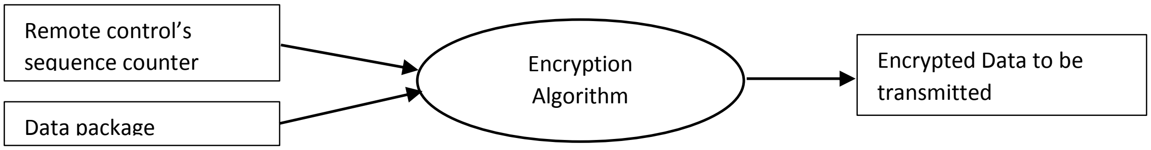
Remote control's sequence counter
Data package
Encryption Algorithm
Encrypted Data to be transmitted

Fig.1. Rolling code encoder in the remote control.

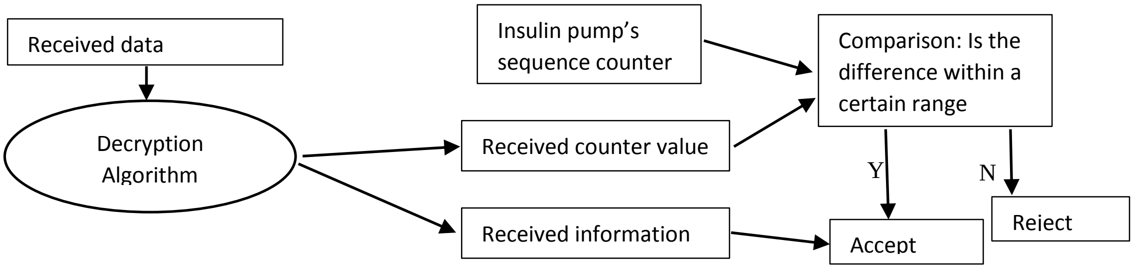

Fig. 2. Rolling code decoder in the insulin pump.

In this report, the remote control's sequence counter and the insulin pump's sequence counter are assumed to be same in regular cases. In view of the original count is from 0 to 255 and repeats after every 256 counts, it's inevitable to cost some space to have 256 keys in circulation. To save some space, 8 keys will be applied, and those key values are stored in the insulin pump system. The remote control sequence counter is the remainder of the value of remote control's sequence counter divided by 8. Meanwhile, to keep the original PIN value to be 36 bits. The following format is applied.

| |
|---|
| Original PIN: 36 bit fixed value. |

| I0<br>(5 bits) | I1<br>(3 bits) | I2<br>(5 bits) | I3<br>(3 bits) | I4<br>(5 bits) | I5<br>(3 bits) | I6<br>(12 bits) |
|---|---|---|---|---|---|---|

Fig. 3. The format of the PIN used to replace the original fixed 36-bit binary sequence.

I0 is 11111 denoting that a new format PIN has been used. The following I1 has 3 bits indicating the value of the calculated remainder which is from 0 to 7. I2 is 5 bits, used to identify if the following mask has been enabled. If its value is 11111, then the following 3 bits indicates the bit value at the certain location of the first 6 bits of counter has been reversed, otherwise the I3 is just random generated numbers. The following I4 has 5 bits as well. The function of I4 is as the same as I3, indicating the second mask has been enabled. The only difference is that the I5 is used to show the marked location of the second 6 bits of the counter value. As for the last 12-bit number is just a random binary variables used to make the new PIN in a format of 36 bits. Thus, the total length of the input data package remains to be 80.

I have neither given or received any unauthorized aid on this assignment.

The detailed process of encryption is shown in Fig. 4. Once the encrypted data is transmitted to the insulin pump system, the value of I1 is used to find the input key value and compared to the PIN in the insulin pump system indicated by its sequence counter. If the received PIN after decryption has no problem, then the input counter value is compared to the insulin system's counter value. If they have a large difference, such abnormal activity will be reported to the phone and trigger the alarm (Fig. 5). Otherwise, the sent data package will be accepted, and the insulin will be injected accordingly. The alarm system in the Matlab is a function that prints an error message to wait for further instruction from the diabetes system's user.

Original input: 0000 110110001001000110010100001110110100 110011111000 001000001111 000000000010 0101

Device type: 0000

Device PIN: 110110001001000110010100001110110100

Information: 110011111000

Counter: 001000 001111 → '8f' = 143

Last four fixed number: 0101

Process:

**I0**: 11111 (enable encryption); **I1**: 111 (because the remainder of 143/8 is 7) ;

**I2**: 11111 (enable the first mask); **I3**: 010 (because the remainder of 7/6 +1 =2);

**I4**: 11111 (enable the second mask); **I5**: 100 (6-I3=4).

As for the last 12 bits are random numbers used to confuse hackers. Thus the encrypted PIN:

11111 111 11111 010 11111 101 0000011010111100

The encrypted input data package:

80 bits

0000 11111 111 11111 010 11111 101 0000011010111100 110011111000 0**1**1000001**0**11 000000000010 0101

Fig. 4. The process of encryption in the remote system.

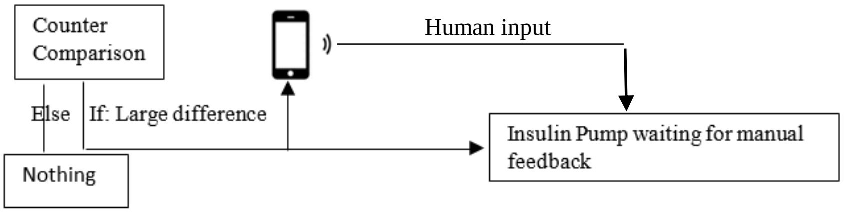

Fig. 5. Alarm system for the abnormal activities.