"Localization Systems for Wireless Sensor Networks", *IEEE Wireless Communications*", PP 6-12. Vol 14 December 2007.
[5] Rui Huang , Gergely V.Zaruba, " Incorporating Data from Multiple Sensors for Localizing nodes in Mobile Ad HocNetworks" , *IEEE Trans.Mobile Computing* , pages1090-1104 , vol.6 , No.9, Sep2007.
[6] L. Hu, D. Evans, "Localization for mobile sensor networks", in: *Tenth International Conference on Mobile Computing and Networking (MobiCom'04)*, Philadelphia, Pennsylvania, USA, September 2004, pp. 45–57.
[7] Rui Huang and Gergely V. Z´aruba,"Location Tracking in Mobile Adhoc Network using particle filter", *Journal of Discrete Algorithms*,vol 5 Issue 3,pp.455-470,2007.
[8] Huynh Quan Hieu,Vu Dinh Thanh, "Ground Mobile Target Tracking By Hidden Markov Model", *Science & Technology Development*, Vol 9, No.12 – 2006.
[9] Weidong Wang. Qingxin Zhu, "Sequential Monte Carlo Localization In Mobile Sensor Networks," *Springer Science+Business Media*, LLC2007,WirelessNetworksDOI 10.1007/S11276-007-0064-3
[10] Dieter Fox, Jeffrey Hightower, Lin Liao,Dirk Schulz,Gaetano Borriello,"Bayesian Filtering for Location Estimation", *Pervasive Computing*, September 2003,pp.24-33.
[11] Babak Pazand, Chris McDonald,"A Critique of Mobility Models for Wireless Network Simulation",*ICIS 2007,IEEE Computer Society*,DOI 0-7695-2841-4/07.
[12] Tracy Camp,Jeff Boleng,Vanessa Davies," A survey of Mobility Models for Ad Hoc Network Research","*Wireless Communication & Mobile Computing(WCWC):special issue on Mobile Ad Hoc Networking:Research,Trends and Applications"*,Vol 2, no. 5, pp. 483-502, 2002

Remote Keyless Entry System with Floating Code

Oleksandr Karpin, Eugene Miyushkovych, Volodymyr Sokil

Computer Engineering Department, Lviv Polytechnic National University, S. Bandery Str., 12, Lviv, 79013, UKRAINE,
E-mail: karpinoo@polynet.lviv.ua, miyushk@polynet.lviv.ua

This paper presents an example of a remote keyless entry system. This system is based on a strong cryptography authentication method and uses a bidirectional Cypress WirelessUSB™ link. It can be used as the base for a complete car alarm or a secure remote keyless entry system (door opener, access control, etc).

Key words – microelectronics, cryptography, authentication, PSoC™, keyless entry.

I. Introduction

Remote keyless entry systems are commonly used for different purposes. Car security systems, garage door control, home security (including room access control), and secure remote controls for devices are just some of the uses of remote keyless entry systems. These systems have different functionality, but use the same type of security components.

Theoretically, any antitheft system can be broken. The alarm system reduces the amount of time that a criminal has to work before being detected. Criminals must know how to swiftly disable the alarm system or use a different method.

Security systems usually consist of two parts; a sensor network and a remote keyless entry system. If the sensor network is correctly designed and mounted so that a criminal does not have physical access to the sensors or the alarm, the keyless entry system becomes the weakest part of the system and the focus of efforts to defeat the system.

The remote keyless entry provides security on two levels; the low level and the high level. The low level is the radio signal transmission. The high level is the secure data transmission channel. The most important part of the secure data transmission channel is the remote control authentication protocol.

II. Radio Channel Security

Most modern alarm systems use the radio channel with band pass modulation, such as amplitude shift keying (ASK), frequency shift keying (FSK), phase shift keying (PSK), or a combination of these. The radio channel uses a narrow-band radio signal spectrum. Using special grabbers these signals can be easily intercepted and stored by criminals. If the alarm system uses a one-way communication link with static code, the criminal can simply replay the stored signal at a later time to gain access to the object.

If the alarm system uses two-way communication and floating code, but does not use reply protection, the criminal can simply transmit all possible reply codes over a period of time to eventually gain access to the object.

Finally, even if the criminals cannot directly use intercepted signals, they can analyze the authentication protocol with the help of the intercepted signals, other obtained data, and powerful computer networks, to eventually learn how to defeat the system.

Therefore, none of these methods using narrow-band radio provides sufficient protection.

The proposed solution to this problem is to use a spread spectrum radio channel. The spread spectrum radio channel system must satisfy three conditions:

- The frequency band must be wider than necessary for data transmission.
- The spectrum spreading must use a coding signal that is not dependant on the data transmitted.
- The receiver must reconstruct the signal using the receiving signal and a synchronized copy of the coding signal.

There are three methods of spectrum spreading; direct sequencing (DS), frequency hopping (FH), and time hopping (TH). A combination of these methods can be used.

We propose to use Cypress WirelessUSB™ radio modems, based on the CYWUSB6934 Radio SoC. This is a highly integrated, low cost radio transceiver, which operates in the unlicensed Industrial, Scientific and Medical (ISM) band (2.4–2.483 GHz). These modems can provide radio link ranges up to 10 meters without an external power amplifier and up to 1 km when an external power amplifier is used. More detailed information on these radio modems can be found in the CYWUSB6932/CYWUSB6934 data sheets [1].

As an alternative to the separate radio modems, the new Cypress Semiconductor CYWUSB6953 WirelessUSB™ PRoC™ (Programmable Radio System-on-Chip) can be used. The CYWUSB6953 WirelessUSB PRoC is the world's first low cost Flash programmable microcontroller with an integrated ISM band radio transceiver. More detailed information on this unique IC can be found in the data sheet [2]. The CYWUSB6953 is the ideal solution for this system.

These modems use the Direct Sequence Spread Spectrum (DSSS) radio channel.

The WUSB modems use a random bit sequence as the coding signal. The coding signal is stored in both the transmitter and receiver. The length of this sequence depends on data bit rate. For example, if the data bit rate is 15,625 kbps, the coding signal length is 64 bits. Because there are 2^{63} variants that a criminal would have to try, a brute force attack would require large amounts of computing power and time.

There are other methods the criminal may use to obtain the coding signal. These methods are too complex to explain in this paper. Nevertheless, if the criminal obtains the coding signal, they still have not defeated the security system. There is a second layer of authentication to prevent these kinds of attacks.

III. Authentication Protocol

If criminals manage to defeat the security provided by the radio channel, they can try to simply reuse the intercepted signals or develop a remote control emulator. For protection against this type of attack, you employ an authentication protocol. The authentication protocol provides a secure channel between the base station and an authorized remote control.

Authentication is the process of determining whether something is what it says it is. Usually the subject of authentication confirms its identity by demonstrating knowledge of some secret information such as a key or password.

One of the most popular and well known methods is simple authentication based on a nonexpendable password. As described previously, this password can be intercepted and reused. An authentication procedure based on the one-time passwords is more effective. For each new transaction, a new password is used. Because one-time passwords are only valid for a single transaction, intercepting the password is useless. It cannot be reused.

One-time password authentication is based on the challenge-handshake protocol. To check the authenticity of the response, the authenticator (A) sends a challenge message to the peer (B). The challenge consists of some unpredictable value, x (a random number, for example). B responds with a value calculated using some function, f(x), known to both A and B. The authenticator checks the response against its own calculation of the expected function value. If the values match, the authenticator can be sure of the identity of the peer, B.

The system detailed in this paper uses a modified challenge-handshake protocol. Since the goal of our work is a low-cost bidirectional alarm system with feedback, the protocol provides two operational modes; normal operation and alarm operation. The flowcharts of these two operational modes are shown in Fig.1 and Fig.2.

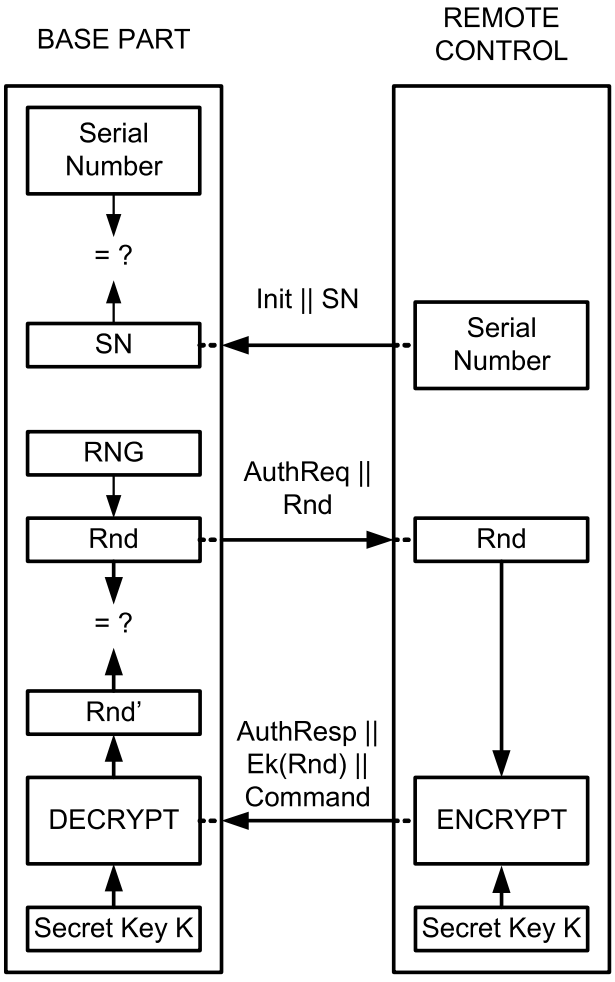

Fig.1 Authentication Protocol – Normal Operation Mode

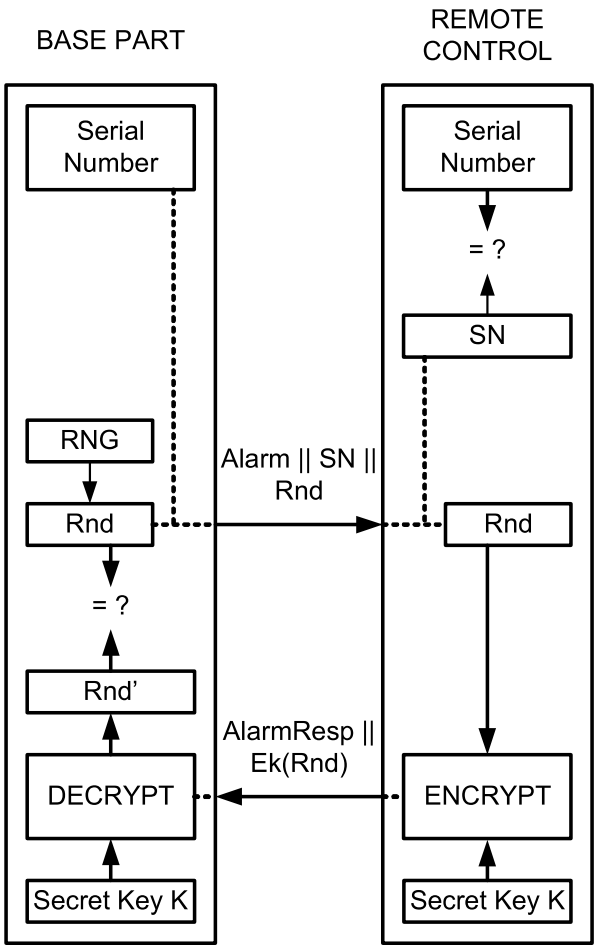

Fig.2 Authentication Protocol – Alarm Operation Mode

In normal operation mode shown in Fig.1, the transaction is initiated by the remote control (RC). This mode is used during normal operation such as locking or unlocking the car by remote control.

The protocol's first step in normal operation mode is to identify the remote control. The RC sends its serial number as an identifier.

If the base recognizes the RC serial number, it generates a random number and sends the authentication request. In this protocol, instead of a function, the remote control encrypts the random number. Encryption is the algorithmic process of obscuring data so there is a low probability of being able to use it without a confidential process or key. The RC encrypts the received random number and sends it back to the base station along with a command ("unlock the doors" for example).

The base station extracts the encrypted random number from the authentication response and decrypts it. If the decrypted and stored values are not equal to each other, the command is ignored.

In alarm operation mode shown in Fig.2, the transaction is initiated by the base station. This mode is used if the sensor network detects a break-in attempt or other alarm event. In this case the alarm signal must be transmitted to the remote control. Because the transaction is initiated by the base station, the RC must identify the base station by the same serial number procedure used previously. The authentication mechanism is the same as in normal operation mode.

The authentication protocol uses two cryptography building blocks; a random number generator and symmetric encryption. This system can use either a true random number generator or a pseudo-random number generator as long as it has a sufficiently long generation period. More detailed information about random and pseudo-random numbers and an example of true random number generator implementation can be found in [3].

The encryption algorithm should have the following characteristics:

- It must be resistant to known cryptanalysis attacks (such as brute force attack, linear and differential cryptanalysis).
- It must be compact so that it does not consume more memory than is available on the chosen Cypress PSoC device.
- It should be relatively fast, though this requirement is not absolutely necessary.

The chosen algorithm for this application is the symmetric block encryption algorithm, RC5. A block encryption algorithm processes a block of plain data as a whole. RC5 is a parameterized algorithm with a variable block size, a variable number of transformation rounds, and a variable key size. The block size depends on word's length. Allowable choices for the block size are 32 bits, 64 bits, and 128 bits (corresponding to machine word lengths of 16, 32, and 64 bits). The number of rounds can range from 0 to 255. The key can range from 0 bits to 2040 bits in size.

Such built-in variability provides efficiency and flexibility at all levels of security. RC5 algorithms are designated RC5-w/r/k, where:

- w is the machine word length in bits,
- r is the number of rounds,
- b is the key size in bytes.

For example, RC5-32/12/5 uses 32-bit words (64-bit block length), 12 rounds, and a 40-bit (5 byte) key. This is what is used in this application.

IV. Prototype System Hardware

To check the performance of the proposed solution, a very simple prototype system was developed. This system consists of two boards: base station and remote control (RC).

The RC prototype board is very simple, but sufficient for testing. It contains only a PSoC® CY8C21534 device, a radio modem connector, a power supply circuit, and a user interface.

The WirelessUSB radio modems are separate, ready-to-use radio modems. The JUNO-L WirelessUSB™ radio modules are produced by Unigen Corporation (www.unigen.com), a Cypress partner.

The user interface is represented by two buttons (for different operations), and one LED, intended as an indicator that an alarm signal was received from the base station. The low-drop linear regulator provides a 3.3V power supply for the all devices' components.

The base station requires more hardware and memory than the remote control, so the PSoC CY8C29466 is used.

Two LEDs indicate when switches are pressed on the RC. One switch is used to simulate an alarm signal event.

V. Prototype System Firmware

The prototype system firmware implements secure radio channel support and general device control. It consists of several modules that serve distinct functions such as data exchange via the radio link between the RC and the base station, authentication protocol, and user interface support.

The data exchange module consists of:

- Low-level subroutines that provide communication with radio modem through the SPI interface
- Subroutines that implement error correction, coding, and decoding algorithms
- A serial transceiver

The first and second parts are described in detail in [4].

The serial receiver control unit state diagram is shown in Fig. 3.

Initially, the receiver is in READY state. The receiver stays at this state until a data packet start byte is received (1). After the start byte is received, the receiver moves to the RECEIVING state (2). This state will be preserved until the packet's stop byte is received (3). When this happens, the receiver returns to READY state (4).

The receiver also can return to the READY state (4) if any of the following events occur:

- A byte is received with an error.
- More than the buffer-size packet bytes are received without a stop byte.
- The timeout expires.

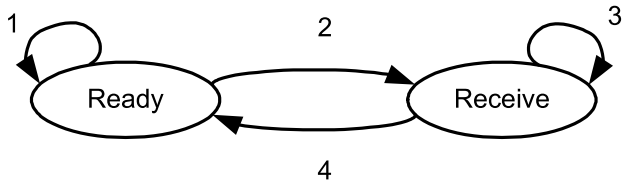

Fig.3 Receiver's Control Unit State Diagram

The transmitter is very simple. At the start of transmission, it sends one preamble byte (0x55) and a start byte (0xAC). After the data packet transmission, a stop byte (0x53) is sent.

The command set shown in Table 1, is used for the authentication protocol.

TABLE 1

AUTHENTICATION MODULE COMMAND SET

Opcode	Command	Fields
1	Init_Command	OP, SN
2	AuthReq_Command	OP, SN, PAS
3	AuthResp_Command	OP, SN, PAS
4	Finish_Command	OP, SN, INF
5	Sync_Command	OP, SN
6	Alarm_Command	OP, SN, PAS
7	AlarmResp_Command	OP, SN, PAS

The behavior of the remote control is defined by the RC control unit, shown in Fig.4. The initial state of the RC's control unit is the IDLE state. The RC remains in IDLE state (1) until any of the following events fire:

- One of two buttons is pressed. This is normal operation mode.
- The alarm command from the base station is received. This is alarm operation mode.

If the user presses a button (for example, "unlock car") the control unit moves to the INIT state (2). In this state the Init_Command is formed and sent to the base station. The control unit goes to the AUTH state (3).

It stays at AUTH state (4) until the AuthReq_Command is received or the timeout expires. If the timeout expires, the control unit goes back to the IDLE state (5). When the AuthReq_command is received, the password from the AuthReq_Command is encrypted and the results are sent with the AuthResp_Command to the base station. The control unit moves to the last state, FINISH (6).

It stays at FINISH state (7) until the Finish_Command is received or timeout is elapsed. After that the control unit returns to the IDLE state (8).

In the alarm operation mode, the control unit moves to the ALARM state (9). The RC transmits the Alarm_Resp command to the base station and goes back to the IDLE state (10).

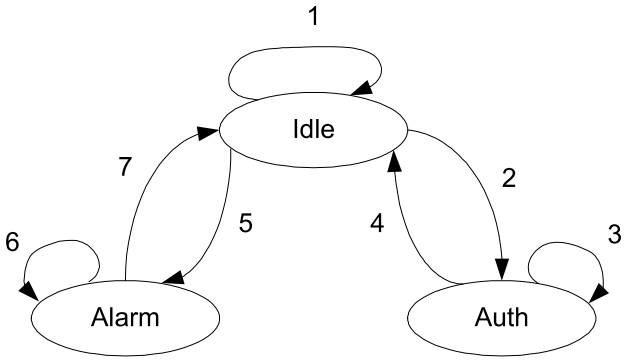

Fig.4 RC's Control Unit State Diagram

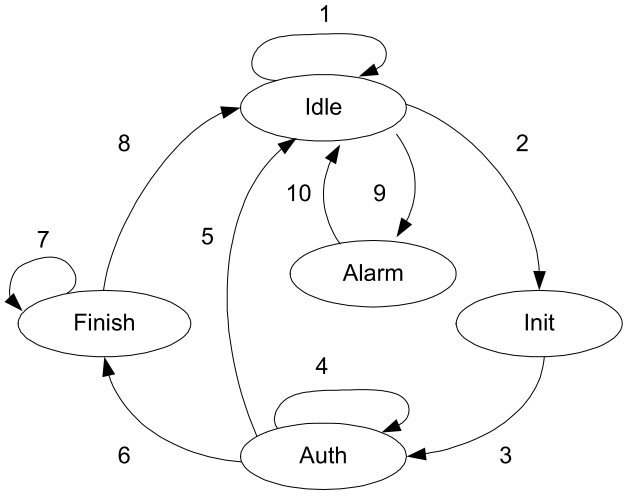

Fig.5 Base Station's Control Unit State Diagram

The base station control unit is shown in Fig.5. The initial state of the base station control unit is the IDLE state. In normal operation mode the base station receives the Init_Command from RC and goes to the AUTH state (2). The AuthReq_Command is formed and sent to the RC.

The control unit stays at this state (3) until the AuthResp_Command is received or the timeout expires. If the timeout expires, the base station goes back to the IDLE state (4). When the AuthResp_Command is received, the authentication check is executed. If the authentication passes, the RC command is executed. After that, the Finish_Command is formed and sent to the RC. The control unit returns to the IDLE state (4).

If the alarm interrupt signal from the sensor's network is detected, the control unit moves to the ALARM state (5). In this state, the Alarm_command is periodically sent to the RC (the period is 1s). This state will be preserved (6) until the AlarmResp_Command from RC is received. After that, the control unit returns to the IDLE state (7).

Conclusion

This paper describes the communication module for an alarm system with a bidirectional interface. This module provides security on two levels. The first level is the interception-proof, noise-immune radio channel using the spread spectrum radio IC transceiver. The second level is the strong authentication for all RC commands. The authentication protocol is based on one-time passwords and a challenge-handshake procedure.

The authentication protocol implementation uses the RC5 encryption algorithm with a 40-bit key to satisfy US export restrictions. But key length can be easily increased up to 128 bits. Moreover, the whole algorithm can be easily replaced by any other customer-designed encryption algorithm with a 64-bit block size.

The functionality of the communication module can easily be modified and expanded. For example, it can be adapted to work with several RCs. To do this you would add a table with authorized RC serial numbers and corresponding secret keys to the firmware of the base station. The general system behavior can be changed so the system can be used in different keyless entry systems.

More detailed information about this project (including schematics and firmware) you can find in [5].

References

[1] Cypress CYWUSB6934 WirelessUSB™ Data Sheet (www.cypress.com)

[2] Cypress CYWUSB6953 WirelessUSB™ PRoC™ Data Sheet (www.cypress.com)

[3] Hardware Random Number Generator – AN2307 (www.cypress.com)

[4] Forward Error Correction using a WirelessUSB Radio System-on-Chip (SoC) Modem – AN2268, (www.cypress.com)

[5] Remote Keyless Entry Car Alarm with Floating Code – AN2308 (www.cypress.com)

Hardware Bitstream Sequence Recognizer

Oleksandr Karpin, Volodymyr Sokil

Computer Department, Lviv Polytechnic National University, S. Bandery Str., 12, Lviv, 79013, UKRAINE,
E-mail: karpinoo@yahoo.com, sokilvm@ukr.net

Abstract *– this paper describes how to implement in hardware a bitstream sequence recognizer using the PSoC™ Pseudo Random Sequence Generator (PRS) User Module. The PRS can be used in digital communication systems with the serial data interface for automatic preamble detection and extraction, control words selection, etc.*

Key words – PSoC, digital communication system, microcontrollers.

I. Introduction

Most often communication systems use a serial interface for data transmission. In such cases, the bit flow must contain some auxiliary service bit sequences. For example, there might be a preamble sequence for receiver adjustment or a synchronization sequence (synchroword) for byte parsing.

The preamble sequences can be extracted both on the firmware or hardware levels, but receiver synchronization can only be performed in hardware. The common and necessary element of these applications is the bitstream sequence recognizer. An application of this type, with 8, 16, 24 or 32 bits, can easy be built on the PSoC™ device PRS User Module.

II. Recognizer implementation

The PRS block diagram is shown in Figure 1. It is a modular linear feedback shift register that generates a pseudo random bit sequence.

The Polynomial register value defines the internal block structure. If this register is initialized to a zero value, the PRS transfers to the Shift register (Figure 2). But this register has one very important feature – the value of this register can be compared with a predefined Seed register value. This is a wonderful base for the bits fragment recognizer.

The Shift register in the standard PRS mode has no data input and output connections. These connections must be defined manually by initialization of the special Control register. In this application, it is only necessary to route a register data input and clock to the internal structure of the target device. The clock source is easily set in PSoC Designer™ Device Editor. The input data source is defined by the PRSxx_x_INPUT_REG (DxBxxIN) register. All possible values of this register are shown in Table 1.