Artur SMOLCZYK[1], Volodymyr KHOMA[1,2], Yaroslav SOVYN[2]

Politechnika Opolska, Wydział Elektrotechniki, Automatyki i Informatyki (1),
Narodowy Uniwersytet "Politechnika Lwowska", Instytut Technologii Komputerowych, Automatyki i Metrologii (2)

doi:10.15199/48.2017.12.61

Bezprzewodowy elektroniczny układ kontroli dostępu do PC

Streszczenie. *W artykule został opisany układ uwierzytelniania, łączący w sobie cechy i funkcje bezstykowego dostępu do PC i klucza elektronicznego USB. Taki układ zapewnia wysoki poziom bezpieczeństwa i elastyczność wykorzystania. Przedstawiono sprzętowo-programową realizację układ uwierzytelniania w oparciu o technologię Bluetooth.*

Abstract*.. In this paper authentication device that combines features and functions of wireless PC access and hardware USB-key is described. This device provides high security level, flexibility and lower price. Device hardware and software design based on Bluetooth technology are overviewed.* ***Device that combines features and functions of wireless PC access and hardware USB-key***

Słowa kluczowe: uwierzytelnianie, dostęp bezprzewodowy do PC, technologie Bluetooth, funkcja haszująca, algorytm HMAC.
Keywords: authentication, wireless PC access, Bluetooth technology, hardware USB-key, hash-function, algorithm HMAC.

Wprowadzenie

Wraz z rozwojem technologii komputerowych, wdrażaniem ich w różne obszary życia i działalności człowieka, wzrasta zapotrzebowanie na zapewnienie bezpieczeństwa informatycznego. Jednym z podstawowych zagadnień w tym kierunku jest zarządzanie dostępem do zasobów informatycznych. Praca każdego systemu kontroli dostępu wymaga uwierzytelniana osób, a sam proces uwierzytelniana może być zrealizowany w jednej z trzech technologii [1,2]:

1) co dana osoba wie, na przykład hasło dostępowe;
2) co dana osoba posiada, na przykład kartę elektroniczną z kodem dostępu;
3) kim dana osoba jest, na przykład odczyt danych biometrycznych takich jak tęczówka oka.

Każda z wymienionych technologii uwierzytelniania ma swoje zalety i wady. I tak uwierzytelnianie w oparciu o wiedzę jest najprostsze w realizacji, ale też nie zapewnia wysokiego poziomu bezpieczeństwa. Z kolei systemy biometryczne mogą zapewnić wysoki poziom bezpieczeństwa, ale są systemami złożonymi. W układzie współrzędnych konkurujących parametrów „złożoność-bezpieczeństwo" systemy uwierzytelniania oparte o karty elektroniczne zajmują pozycję kompromisową (pośrednią), zapewniając średni poziom zabezpieczeń oraz koszty realizacji.

Obecnie w postaci środków uwierzytelniania elektronicznego mogą występować klucze kontaktowe TouchMemory, karty chipowe (ang. smart-card), karty magnetyczne, karty zbliżeniowe lub bezstykowe (RFID-card, proximity-card), klucze sprzętowe USB [3, 4, 5].

Analiza ostatnich rozwiązań i cel artykułu

Obecnie wśród środków elektronicznych systemu kontroli dostępu do PC największą funkcjonalność oraz najwyższy poziom bezpieczeństwa zapewniają chipowe karty inteligentne i klucze USB [6, 7, 8].

Karta inteligentna jest to karta plastikowa z wbudowanym mikrosterownikiem, który pełni funkcje ograniczania dostępu do przechowywanej w pamięci informacji, dokonuje obróbki i wymiany danych oraz realizuje algorytmy kryptograficzne. W wewnętrznej pamięci EEPROM mikrosterownika są przechowywane dane osobowe użytkownika: hasła, klucze, certyfikaty itp. [7, 9, 10]. Istnieją karty inteligentne kontaktowe lub bezstykowe (zbliżeniowe). W obu przypadkach wymagane są specjalne czytniki. Dość wysoka cena czytników i samych kart inteligentnych (rzędu kilkuset USD [10, 11]), póki co, ogranicza ich szerokie wykorzystanie jako elektronicznych identyfikatorów dostępu do komputerów PC.

Systemy uwierzytelniania i ochrony informacji na bazie kluczy USB znalazły szerokie zastosowanie w różnych dziedzinach gdzie są wykorzystane komputery. Taki system nie potrzebuje czytnika, ponieważ wszystkie nowoczesne komputery są wyposażone w porty USB, co w dużym stopniu obniża koszty implementacji systemu kontroli dostępu. Najczęściej wykorzystywane są identyfikatory USB – osobiste układy uwierzytelniania i przechowywania danych, podtrzymujące na poziomie sprzętowym współpracę z certyfikatami cyfrowymi i elektronicznym podpisem cyfrowym. Pozornie taki układ ma postać nośnika flash typu USB Pen Drive, ale zakres wykonywanych funkcji odpowiada karcie inteligentnej [8].

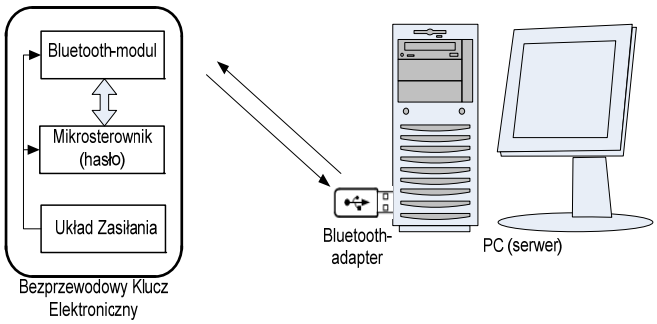
Używanie kluczy USB jest bardzo wygodnie, ponieważ użytkownik nie musi zapamiętywać różnych haseł i kodów dostępu, a wszelka informacja jest przechowywana w samym urządzeniu. Ponadto na tym nośniku mogą być zapisane certyfikaty i podpisy cyfrowe oraz inne poufne dane, przechowywanie których na otwartych nośnikach jest zakazane ze względu na bezpieczeństwo.

Ze względu na swą architekturę wewnętrzną klucze USB mogą być realizowane jako elektroniczne karty procesorowe lub na bazie zabezpieczonych mikrosterowników [8]. W drugim przypadku podstawowym elementem jest procesor, który w całości pełni funkcję klucza, w tym realizuje interfejs USB [12]. Taki układ zawiera moduł pamięci (Firmware Memory) z programem i danymi konfiguracyjnymi oraz moduł pamięci RAM. Z punktu widzenia uwierzytelniania klucz USB musi poddać się identyfikacji wobec PC, jeśli docelowe dane poufne są ulokowane bezpośrednio na komputerze lub serwerze. Sama procedura identyfikacji przebiega w oparciu o jednostronne funkcje haszujące.

W artykule została przedstawiona możliwość realizacji układu uwierzytelniania łączącego w sobie funkcje bezstykowego dostępu i klucza elektronicznego USB, co zapewnia wysoki poziom bezpieczeństwa i elastyczność wykorzystania. Łączny koszt elementów proponowanego urządzenia jest rzędu kilkunastu USD.

Opis części sprzętowej

Dla realizacji bezstykowego układu uwierzytelniania elektronicznego został wybrany interfejs bezprzewodowy Bluetooth, ponieważ jest to rozpowszechniony tani interfejs bezstykowy komputerów osobistych. Schemat strukturalny identyfikatora ilustruje rys. 1.



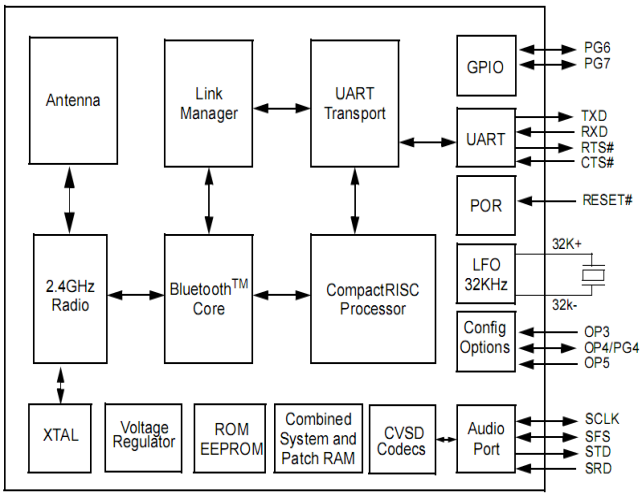
Rys. 1. Schemat strukturalny bezprzewodowego elektronicznego układu kontroli dostępu do PC

Bezprzewodowy elektroniczny układ uwierzytelniający (zwany dalej bezprzewodowym kluczem elektronicznym) został wykonany w postaci breloczka (ang. key ring) i składa się z modułu Bluetooth, mikrosterownika i bloku zasilania. Mikrosterownik wykonuje dostrajanie pracy całego układu, przeprowadza transmisję danych i realizuje kryptograficzny protokół uwierzytelniania. W pamięci EEPROM mikrosterownika jest przechowywany unikalny klucz połączenia (ang. link key), który uczestniczy w algorytmie uwierzytelniania. Komputer powinien być wyposażony w adapter Bluetooth (wbudowany lub podłączony do gniazda USB).

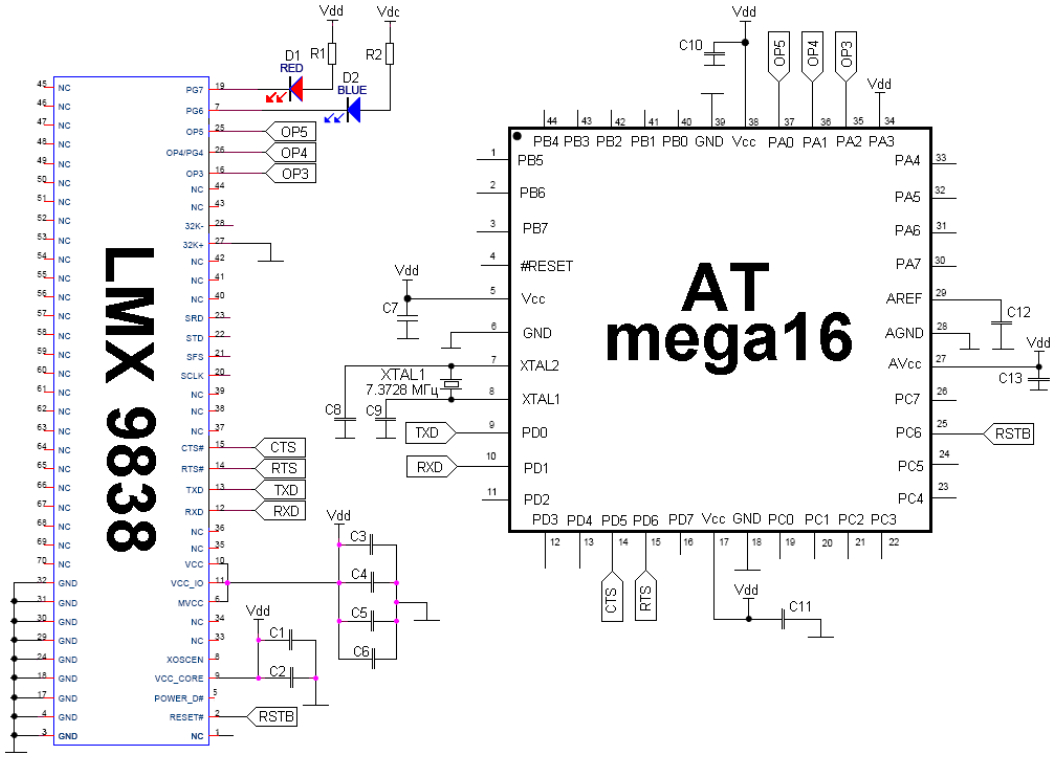
Mimo, że specyfikacja technologii Bluetooth przewiduje przeprowadzenie procedury uwierzytelniania oraz szyfrowanie transmisji, szereg wykrytych podatności w protokole Bluetooth wymaga wprowadzenia dodatkowej procedury uwierzytelniania oprócz istniejących mechanizmów bezpieczeństwa.

Program zainstalowany na PC co sekundę nawiązuje połączenie z bezprzewodowym kluczem elektronicznym i przeprowadza uwierzytelnianie. W razie jego niepowodzenia lub w przypadku niemożliwości nawiązania połączenia przez Bluetooth praca PC zostaje zablokowana. Ponieważ zasięg adapterów Bluetooth wynosi 7-10 metrów (bez zakłóceń), blokowanie będzie dokonywane automatycznie, jeżeli tylko użytkownik oddali się od swojego stanowiska pracy na zbyt dużą odległość. W trakcie inicjalizacji połączenia urządzenie Bluetooth jest jednoznacznie identyfikowane na podstawie adresu fizycznego (6-bajtowy MAC). Dzięki temu system działa prawidłowo niezależnie od liczby urządzeń Bluetooth będących w zasięgu hosta.

Moduł Bluetooth został zrealizowany w oparciu o układ scalony LMX9838 firmy Texas Instruments [13]. Jest to w pełni zintegrowany (10×17×2 mm) moduł zawierający wbudowaną antenę dla częstotliwości 2,4 GHz, 16-bitowy sterownik RISC, interfejs uniwersalnego asynchronicznego odbiornika-nadajnika (UART) dla wymiany danych z sterownikiem (rys. 2).



Rys. 2. Schemat strukturalny modułu Bluetooth LMX9838



Rys. 3. Schemat połączenia modułu Bluetooth LMX9838 z sterownikiem

Moduł realizuje takie profile Bluetooth jak: GAP (Generic Access Profile), SDAP (Service Discovery Application Profile) i SPP (Serial Port Profile), które są potrzebne dla pracy układu uwierzytelniania. Szybkość wymiany danych w profilu SPP jest określana przez sterownik (maksymalna wartość wynosi 921600 bit/s) [13].

Do realizacji sterownika został wybrany mikrosterownik ATmega16 firmy Atmel. Na wybór tego mikrosterownika wpłynęła dość duża objętość pamięci (Flash – 16 kB, SRAM – 1 kB) oraz wysoka skuteczność jądra AVR (1 MIPS/MHz). Ważna jest również obecność we wszystkich mikrosterownikach rodziny AVR specjalnych komórek (Lock Bits), dla ochrony danych przed odczytem/zapisem zawartości pamięci FLASH programów i pamięci EEPROM [14,15].

Fragment schematu układu uwierzytelniania ilustrujący połączenie sterownika z modułem Bluetooth LMX9838 jest pokazany na rys. 3.

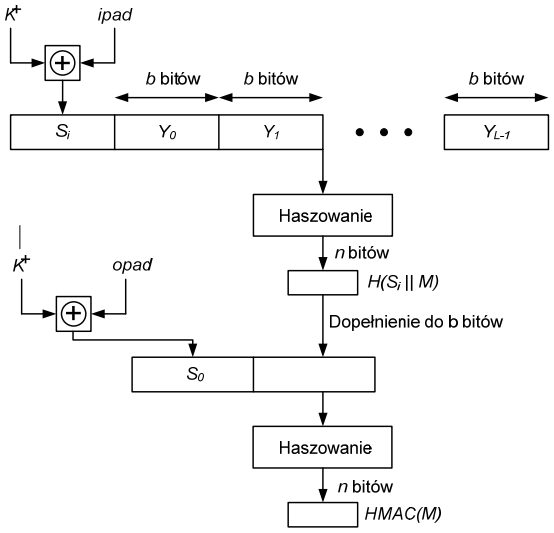
Na wejściach *OP3-OP5* mikrosterownik zadaje szybkość wymiany danych z LMX9838 przez linie RxD/TxD. Transmisja danych i instrukcji jest przeprowadzana w trybie pakietowym, linie CTS/RTS są przeznaczone dla dopasowania szybkości mikrosterownika i modułu Bluetooth. Przez wejście RSTB mikrosterownik resetuje moduł Bluetooth.

Protokół uwierzytelniania

W celu uwierzytelniania PC jako serwer po wykryciu bezprzewodowego klucza elektronicznego (klucza Bluetooth) nadaje mu pewną wiadomość (losową liczbę), generowaną technologią haseł jednorazowych. Po otrzymaniu tej wiadomości bezprzewodowy klucz elektroniczny wykonuje podpis jej wartości hasz w oparciu o poufny kod i zwraca wynik do komputera.

Program uwierzytelniania na PC przeprowadza analogiczne obliczenia hasz i porównuje rezultaty. Jeżeli skróty są zbieżne, to proces uwierzytelniania jest postrzegany jako udany. Nie jest możliwe uzyskanie właściwego wyniku haszowania bez znajomości poufnego kodu, który nie opuszcza granic urządzenia [16].

Jako protokół uwierzytelniania został wybrany popularny standardowy algorytm HMAC (The Keyed-Hash Message Authentication Code) [16]. Na rys. 4 jest pokazany ogólny schemat algorytmu HMAC.



Rys. 4. Etapy działania algorytmu HMAC

Przyjęto następujące oznaczenia:

H – wbudowana funkcja haszująca;

M – wiadomość podawana na wejście HMAC (wraz z bitami wypełniacza);

Yi – *i*-ty blok wiadomości *M*, 0<*i*<*L*-1;

L – liczba bloków wiadomości *M*;

B – liczba bitów w pojedynczym bloku;

n – długość kodu hasz generowanego wbudowaną funkcją haszującą *H*;

K – klucz tajny; jeśli długość klucza jest większa od długości bloku *b*, klucz podaje się na wejście funkcji haszującej, żeby otrzymać *n*-bitowy klucz;

K+ – klucz *K* z dopisanymi na początek zerami, żeby jego długość wynosiła *b* bitów;

ipad – wartość 00110110 (36h), powtórzona b/8 razy,

opad – wartość 01011010 (5Ch), powtórzona b/8 razy.

Algorytm można opisać następująco:

1. Do wartości *K* od lewej dodaje się zera, aby otrzymać b-bitowy wiersz *K*+.

2. Wylicza się alternatywę wykluczająca (XOR) wartości *K*+ oraz ipad, otrzymując w wyniku b-bitowy blok *Si*.

3. Do *Si* dodaje się wiadomość *M*.

4. Do sekwencji otrzymanej w p. 3, stosuje się funkcję haszującą *H*.

5. Wartość *K*+ jest „XORowana” z *opad*, tworząc *b*-bitowy blok S_0.

6. Wynik haszowania otrzymany w p. 4, dołącza się do S_0.

7. Do sekwencji otrzymanej w p. 6, stosuje się funkcję haszującą *H*, a wynik podaje się na wyjście algorytmu.

Algorytm HMAC można zatem zapisać wzorem:

$$HMACK = H[(K+ opad) \,||\, H[(K+ ipad) \,||\, M]].$$

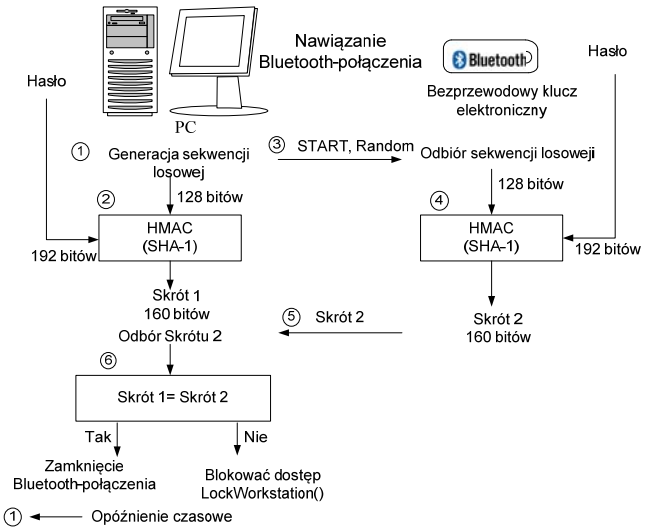
Do zalet HMAC można zaliczyć:

- możliwość wykorzystywania kluczy tajnych dla wzmocnienia uwierzytelniania;
- możliwość wykorzystywania bez modyfikacji istniejących funkcji haszujących;
- możliwość zamiany funkcji haszującej, na przykład w przypadku jej złamania;
- wysoka wydajność algorytmu.

Jak już podkreślono wyżej, algorytm HMAC nie definiuje konkretnej funkcji haszującej *H*, która ma być wykorzystana. Przy opracowywaniu identyfikatora elektronicznego został wybrany algorytm SHA-1 głównie ze względu na fakt, że jest on standardem [17, 18].

W algorytmie SHA-1 rozmiar wejściowego bloku *b* wynosi 512 bitów, a wynikowego kodu hasz *n* – 160 bitów. Długość tajnego klucza *K* ustalono na 192 bity, a dowolne wybranej wiadomości na *M* – 128 bitów.

Na rys. 5 przedstawiono dokładniejszy schemat procedury uwierzytelniania.



Rys. 5. Schemat uwierzytelniania w oparciu o bezprzewodowy elektroniczny układ kontroli dostępu do PC

Opis części programowej

Oprogramowanie składa się z dwóch części – programu klienckiego dla mikrosterownika oraz programu dla serwera zainstalowanego na PC. Program dla mikrosterownika Atmega16 został napisany w języku C z wykorzystywaniem kompilatora CodeVisionAVR. Obliczanie wartości HMAC zajmuje 766300 takty, co przy częstości zegara 7,3728 MHz wynosi 104 ms.

Serwerowa część programu została napisana w języku C++ w środowisku Microsoft Visual Studio. Do realizacji funkcji kryptograficznych wybrano bibliotekę CryptoAPI jako składnik systemów operacyjnych rodziny Windows. W środowisku CryptoAPI realizacja wszystkich algorytmów (szyfrowanie, haszowanie, generacja liczb losowych itp.) całkowicie odbywa się nie w samej bibliotece, a w oddzielnych, niezależnych i dynamicznie załadowywanych bibliotekach – tak zwanych „kryptoprowajderach" (Cryptographic Service Provider – CSP). Sama biblioteka CryptoAPI daje końcowemu użytkownikowi zunifikowany interfejs do pracy z CSP [19].

Nie jest możliwe wpłynięcie na przebieg algorytmu realizowanego w kryptoprowajderze, ponieważ komponenty kryptosystemu Windows (wszystkie bez wyjątku) muszą mieć podpis cyfrowy (czyli jest podpisany i odpowiedni plik *dll* kryptoprowajdera). Podane rozwiązanie jest bezpieczniejsze w porównaniu z napisanym osobiście oprogramowaniem realizującym wymienione algorytmy kryptograficzne.

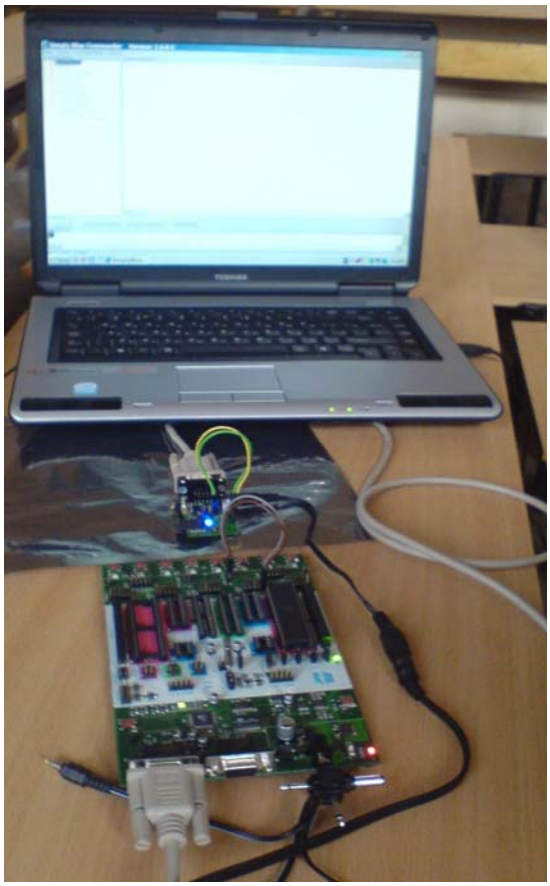
Fragment programu obliczającego wartości HMAC z wykorzystywaniem CryptoAPI jest pokazany na rys. 6 (w celu uproszczenia nie pokazano przerwań funkcji oraz innych mniej istotnych fragmentów kodu).

```
#include <wincrypt.h>
.....
BYTE KeyBlob[] =
{ 0x08, 0x02, 0x00, 0x00, 0x03, 0x66, 0x00, 0x00, // BLOB header
  0x18, 0x00, 0x00, 0x00,                         // Key length, in bytes
  '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0',   // Secret 192 bits
  '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0' };

for(BYTE ii = 0; ii < 24; ii++)  KeyBlob[ii + 12] = Secret_key[ii];

HCRYPTPROV  hProv;
HCRYPTKEY   hKey;

CryptAcquireContext(&hProv,  NULL,  MS_ENHANCED_PROV,
 PROV_RSA_FULL, CRYPT_VERIFYCONTEXT);

CryptImportKey(hProv,  KeyBlob,  sizeof(KeyBlob),  0,
 CRYPT_EXPORTABLE, &hKey);

HCRYPTHASH hHash;
CryptCreateHash(hProv, CALG_HMAC, hKey, 0, &hHash);

HMAC_INFO hmac_i;
hmac_i.HashAlgid = CALG_SHA1;

CryptSetHashParam(hHash, HP_HMAC_INFO, (const BYTE*)
&hmac_i, 0) ;

BYTE Rand_Message[16];
CryptGenRandom(hProv, 16, Rand_Message);

CryptHashData(hHash, Rand_Message, 16, 0);

DWORD count = 0;
CryptGetHashParam(hHash, HP_HASHVAL, NULL, &count, 0) ;

char* hash_value = static_cast<char*>(malloc(count + 1));
ZeroMemory(hash_value, count + 1);

CryptGetHashParam(hHash, HP_HASHVAL, (BYTE*)hash_value,
&count, 0);

for ( BYTE ii = 0; ii < 20; ii++ )  HMAC_digest[ii] = hash_value[ii];
```

Rys. 6. Realizacja funkcji kryptograficznych na bazie CryptoAPI dla programu serwera

Dla realizacji funkcji Bluetooth autorzy wybrali darmowy zbiór protokołów Software Development Kit BCM1000-BTW firmy Broadcom. W skład oprogramowania BCM1000-BTW wchodzą biblioteki dll oraz interfejs do wykorzystania ich w C++. Wymiana dowolnych informacji pomiędzy adapterem Bluetooth komputera a bezprzewodowym kluczem elektronicznym realizowana jest przez profil SPP, który w istocie sprawy jest wirtualnym interfejsem szeregowym COM.

Opisana w artykule koncepcja budowy bezprzewodowego elektronicznego klucza kontroli dostępu do PC została zrealizowana w postaci układu pokazanego na rys. 7 (w realizacji praktycznej rozmiary układu mogą być znacznie mniejsze pozwalając na umieszczenie go w breloku).



Rys. 7. Zdjęcie bezprzewodowego systemu kontroli dostępu do PC z układem debugowania

Na jednej płycie drukowanej umieszczono mikrosterowniik ATmega16 i rezonator krzemowy, a na drugiej - blok debugowania LMX9838Donge oraz moduł Bluetooth LMX9838. Wymiana danych pomiędzy płytami dokonuje się przez gniazdo RS-232. Jako adapter Bluetooth został wykorzystany układ DBT-122 firmy D-Link. Komputer pracuje pod kontrolą systemu operacyjnego Windows 10.

Testowanie oraz eksploatacja bezprzewodowego elektronicznego układu kontroli dostępu do PC wykazały poprawność opisanej koncepcji.

Podsumowanie

Opracowany oraz zrealizowany fizycznie układ bezprzewodowego klucza elektronicznego jest wygodnym rozwiązaniem przeprowadzania kontroli dostępu do komputerów PC. Do zalet proponowanego układu uwierzytelniania należy elastyczność wykorzystania,

ponieważ komunikacja bezprzewodowego klucza elektronicznego a komputera jest oparta o standardowy protokół Bluetooth. Ponadto dzięki wykorzystaniu w urządzeniu algorytmu HMAC jest zapewniany wysoki poziom bezpieczeństwa.

Zasoby mikrosterownika zostały wykorzystane częściowo. Program zajmuje 3086 B (z dostępnych 16384 B), wykorzystuje 54 B (z 1024 B) RAM oraz 32 B (z 512 B) EEPROM dla przechowywania klucza użytkownika. Pozwoli to w przyszłości rozszerzyć możliwości funkcjonalne układu o realizację dodatkowych operacji szyfrowania i deszyfrowania plików oraz bezpiecznego przechowywania danych (przy wykorzystaniu układu pamięci flesz NAND).

***Autorzy**: Dr inż. Artur Smolczyk, Politechnika Opolska, Instytut Informatyki, E-mail: a.smolczyk@po.opole.pl; Prof., dr hab. inż. Volodymyr Khoma,* Politechnika Opolska, Instytut Automatyki, *E-mail: v.khoma@po.opole.pl; Yaroslav Sovyn, dr inż.,* Narodowy Uniwersytet „Politechnika Lwowska", *E-mail: ysovyn@gmail.com*

LITERATURA

[1] Richard E. Smith. Authentication: From Passwords to Public Keys. *Addison-Wesley Professional*, (2002), 576.
[2] Gang Ma , Kehe Wu , Tong Zhang , Wei Li. A Flexible Policy-Based Access Control Model for Workflow, *Przegląd Elektrotechniczny*, Nr 3b, 2012, 67-71.
[3] ISO/IEC 14443-1:2008 Identification cards - Contactless integrated circuit cards - Proximity cards - Part 1: Physical characteristics.
[4] Norman T.L. Electronic Access Control. *Elsevier Science*, (2011), 456.
[5] Bohler L., Daniol M., Wehrle C. Identification of instruments and implants with RFID and Data Matrix Codes for the use at the instrument table. *Przegląd Elektrotechniczny*, Nr 11, 2016, 225-228.
[6] Allen Chang, Han-Chen Huang, Dwen-Ren Tsai, Development of Practical Smart House Scenario Control System, *Przegląd Elektrotechniczny*, Nr 1b/2013, 159-161.
[7] Smart Card Handbook / Rankl W., Effing W. – West Sussex: *John Wiley & Sons, Ltd*, (2003). - 1088.
[8] Dshhunyan V.L., Shangin V.F. Elektronnaya identyfikaciya. Bezkontaktnyye elektronnye identifikatory i smart-karty. M: *OOO „Izdatelsvo AST"*, (2004), 695. (In Russian)
[9] Smart Card Applications. Design Models for using and programming smart cards / Rankl W. – West Sussex: *John Wiley & Sons, Ltd*, (2007). - 217.
[10] Nazimek P., Inżynieria programowania kart inteligentnych. *Wyd-wa Politechnika Warszawska*, Warszawa 2005, 171.
[11] Leszek P., Smart cards - krzemowa inteligencja, PCWorld, 2006: http://www.pcworld.pl/news/Smart-cards-krzemowa-inteligencja,301457.html
[12] Andrzej Bień, Jacek Augustyn: Właściwości czasowe interfejsu USB we wbudowanych systemach pomiarowo-sterujących. *Przegląd Elektrotechniczny*, Nr 7, 2009, 1-7.
[13] LMX9838 Bluetooth Serial Port Module. Data Sheet. *Texas Instruments SNOSAZ9F. Revised December* (2014). http://www.ti.com/lit/ds/symlink/lmx9838.pdf
[14] Jarosław Doliński, Mikrokontrolery AVR w praktyce. *Wydawnictwo BTC*, Warszawa (2004), 452.
[15] Krzysztof A. Wpływ niskich temperatur na czas wykonania operacji w systemach pomiarowych z mikrokontrolerem ATmega16A. *Przegląd Elektrotechniczny*, Nr 11, 2015, 309-312.
[16] FIPS 198-1. National Institute of Standards and Technology. The Keyed-Hasz Message Authentication Code (HMAC). *Federal Information Processing Standards Publication* 198-1, July (2008).
[17] ISO/IEC 10118-3:2004: Information technology - Security techniques - Hash-functions - Part 3: Dedicated hash-functions.
[18] Schneier B. Kryptografia dla praktyków. Protokoły, algorytmy i programy źródłowe w języku C. *WNT Wydawnictwa Naukowo-Techniczne*, 2002, 899.
[19] Scherbakov L.Y. Domashev A.V. Prikladnaya kriptografiya. Ispolzovaniye i sintez kriptpgraficheskih interfeisov. M: *Izdatelsko-torgovyi dom "Russkaya Redakciya*. 2003, 416. (In Russian)