Biuletyn WAT
Vol. LXIV, Nr 4, 2015

Bezpieczeństwo transmisji danych w przemysłowym systemie sterowania

[1]MARCIN BEDNAREK, TADEUSZ DĄBROWSKI

[1]Politechnika Rzeszowska, Wydział Elektrotechniki i Informatyki,
35-959 Rzeszów, ul. Powstańców 12, bednarek@prz.rzeszow.pl
Wojskowa Akademia Techniczna, Wydział Elektroniki,
00-908 Warszawa, ul. gen. S. Kaliskiego 2, tadeusz.dabrowski@wat.edu.pl

Streszczenie. Artykuł poświęcony jest propozycji systemu zabezpieczenia transmisji danych pomiędzy stacjami przemysłowego systemu sterowania. Opisano możliwe warianty zabezpieczenia komunikacji pomiędzy stacjami procesowymi oraz pomiędzy stacją procesową i operatorską. Mechanizm zabezpieczenia transmisji bazuje na algorytmach szyfrowania symetrycznego i asymetrycznego. Proces uwierzytelniania wykorzystuje token programowy i algorytm obliczania jednokierunkowej funkcji skrótu. Podano schemat nawiązania zabezpieczonego połączenia pomiędzy stacjami, w tym etapu uwierzytelniania stacji i szyfrowania przesyłu danych. Proces zabezpieczenia transmisji składa się z czterech podprocesów: (I) uwierzytelniania; (II) transmisji kluczy asymetrycznych publicznych; (III) transmisji klucza symetrycznego; (IV) transmisji danych. Przedstawiony proces zabezpieczania transmisji został zrealizowany w sterowniku przemysłowym oraz w jego emulatorze. Wykorzystano do tego celu języki programowania zgodne z normą PN-EN 61131. Funkcje zostały zaimplementowane w formie bloków funkcyjnych użytkownika umożliwiających zawarcie w strukturze bloku fragmentów mieszanego kodu (w języku ST i FBD) i podzielonych na sześć kategorii: obsługa szyfrowania asymetrycznego; pomocnicze funkcje szyfrowania asymetrycznego; obsługa szyfrowania symetrycznego; pomocnicze funkcje szyfrowania symetrycznego; obsługa obliczania wartości funkcji skrótu; pomocnicze funkcje konwersji.
Słowa kluczowe: bezpieczeństwo transmisji, szyfrowanie, uwierzytelnianie, rozproszony system sterowania
DOI: 10.5604/12345865.1186229

1. Wstęp

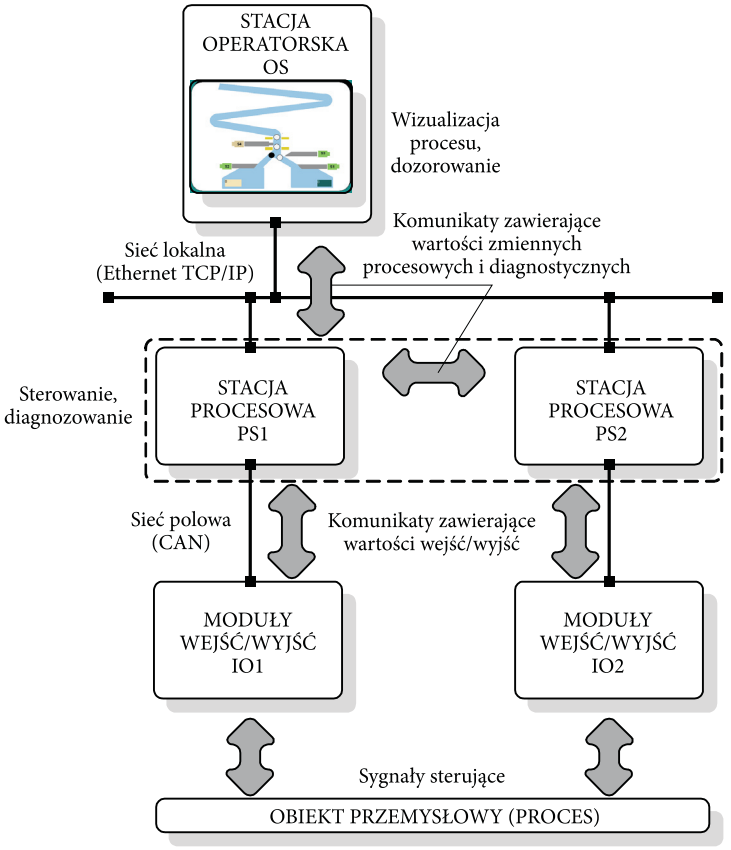
Rozpatrywany w artykule przemysłowy system sterowania składa się przede wszystkim ze stacji procesowych (ang. *Process Station* — PS1 i PS2) [1]. Zadaniem

takich stacji jest sterowanie procesem. Stacje wymieniają pomiędzy sobą komunikaty zawierające wartości zmiennych procesowych i diagnostycznych (rys. 1). Dodatkowo w systemie może występować stacja operatorska (ang. *Operator Station* — OS) prowadząca wizualizację i dozorowanie procesu. Ważnym aspektem komunikacji jest zabezpieczenie transmitowanych danych przed nieuprawnionym dostępem, modyfikacją lub podszyciem się intruza pod jedną ze stacji (ang. *security*) [2]. W systemie występuje kilka poziomów procesów komunikacyjnych:

- *Pomiędzy modułami wejść/wyjść a wyjściami/wejściami obiektowymi*. Wystąpienie niezdatności systemu spowodowane błędami transmisji wiąże się tu tylko z niezdatnością jednego z elementów toru transmisyjnego.
- *Pomiędzy stacją procesową a modułami wejść/wyjść*. Do komunikacji wykorzystywany jest protokół CAN. Bezpieczeństwo transmisji opartej na standardzie CAN rozpatrywane jest m.in. w [3, 4]. Jest ona jednak prowadzona na niewielką odległość oraz w miejscach o znikomej podatności na oddziaływanie czynników destrukcyjnych powodowanych przez intruza.
- *Pomiędzy stacjami procesowymi oraz pomiędzy stacją procesową i operatorską*. W tym miejscu występuje kilka możliwych wariantów zestawienia połączeń. Komunikacja realizowana jest przy użyciu specjalnie wydzielonej do tego celu lokalnej sieci komputerowej (LAN) [5] z zastosowaniem rodziny protokołów TCP/IP [6]. Podatność na destrukcyjne oddziaływanie intruza jest w tym miejscu znacznie większa, biorąc pod uwagę:
  a) wykorzystanie ustandaryzowanego protokołu komunikacyjnego [3, 4] oraz dostępność dla intruza okablowania sieci LAN;
  b) zastąpienie fragmentów sieci przewodowej przesyłem radiowym [7, 8].

Istnieje kilka rozwiązań realizacji komunikacji pomiędzy stacjami procesowymi prowadzonej wg protokołu TCP/IP z wykorzystaniem standardu Ethernet (wymieniony tu „repertuar" rozwiązań nie kończy listy dostępnych możliwości):

- umieszczenie zadań stacji procesowych we wspólnym drzewie projektu programu sterującego i przesył danych jako wewnętrznych zmiennych systemowych [1];
- wykorzystanie protokołu jednego z dostępnych standardów protokołów przemysłowych, np. Modbus TCP [9];
- zastosowanie serwera OPC i wykorzystanie przez stacje procesowe wartości zmiennych uzyskanych po podłączeniu do serwera przy pomocy wbudowanych mechanizmów klienta [10];
- użycie gotowych bloków komunikacyjnych komunikacji na poziomie TCP/IP dostępnych z poziomu standardowych bibliotek bloków. Pozwala to na zaprojektowanie własnego rozwiązania zabezpieczenia przesyłanych wartości zmiennych, zanim trafią do wejścia bloku wysyłającego. Rozwiązanie to przyjęto jako bazowe do dalszych rozważań.



Rys. 1. Komunikacja w rozpatrywanym systemie

Standardowa komunikacja pomiędzy stacjami odbywa się niezabezpieczonym kanałem transmisyjnym. W artykule przedstawione są elementy zabezpieczenia transmisji pomiędzy stacjami procesowymi. Proponowane rozwiązanie, tj. opracowanie procedur zabezpieczających, bazujących na sprawdzonych i uznanych za obliczeniowo bezpieczne mechanizmach szyfrujących, pozwala uniezależnić się od zamkniętych rozwiązań firmowych.

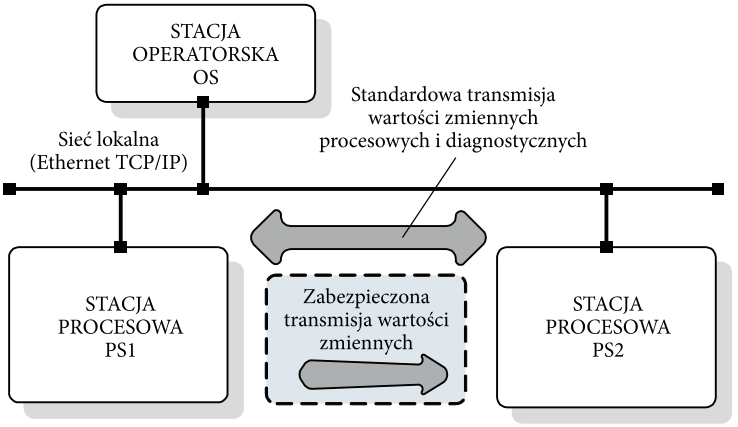
2. Komunikacja zabezpieczona pomiędzy stacjami procesowymi

Spośród dostępnych rozwiązań zabezpieczenia transmisji danych pomiędzy stacjami (zarówno procesowymi, jak i pomiędzy procesową i operatorską) można zastosować:

— elementy pośredniczące (bramy) tworzące pomiędzy urządzeniami wirtualny, szyfrowany kanał (wada: konieczność użycia dodatkowych urządzeń);

— połączenie z OPC z wymuszeniem szyfrowania (wada: nie wszystkie implementacje serwerów/klientów OPC na to pozwalają);
— standardowe bloki przesyłu i opracowane procedury zabezpieczające transmisję wykorzystujące mechanizmy uwierzytelniania, szyfrowania, zapewnienia integralności przesyłanych danych.

Standardową procedurą konfiguracji komunikacji jest zestawienie dla każdej przesyłanej wartości zmiennej pary bloków: nadawczego i odbiorczego [11]. Oprócz występujących w systemie standardowych wymian komunikatów zmodyfikowany proces transmisji danych zorganizowany jest w taki sposób, aby wykorzystując wspomniane pary bloków nadawczo-odbiorczych, uzupełnić go procedurami bezpiecznościowymi. Należy odpowiednio sterować transmisją oraz dokonać szyfrowania danych przed wysłaniem (przed podaniem wartości przesyłanej zmiennej na wejście bloku wysyłającego). Tym samym, obok przesyłu informacji standardowymi kanałami jawnymi, funkcjonuje kilka kanałów transmisyjnych służących do wysyłania zabezpieczonych danych (rys. 2).



Rys. 2. Transmisja danych pomiędzy stacjami procesowymi

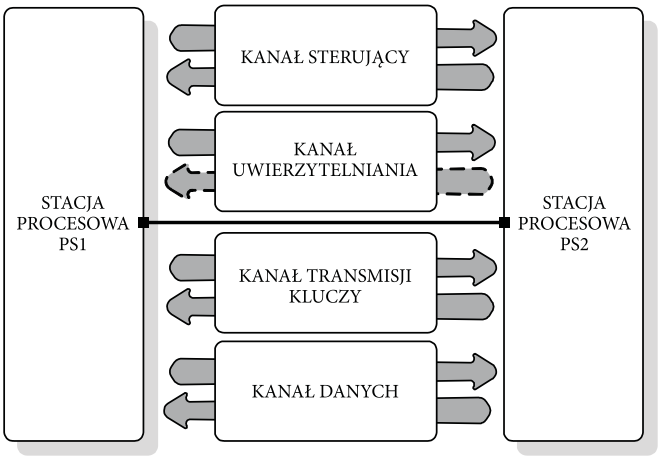
Opracowane procedury zabezpieczające komunikację pomiędzy stacjami procesowymi obejmują kolejno procesy:
— uwierzytelniania stacji procesowej;
— szyfrowania asymetrycznego;
— szyfrowania symetrycznego;
— zapewnienia integralności przesyłanych danych [12, 13].

3. Proces zabezpieczania transmisji

Do poprawnego działania systemu zabezpieczenia transmisji wykorzystuje się cztery kanały komunikacyjne. Są to dwukierunkowe kanały transmisyjne:

— sterujący;
— uwierzytelniania;
— transmisji kluczy;
— danych.

Na rysunku 3 jedna ze strzałek kanału uwierzytelniania jest zaznaczona linią przerywaną, w celu podkreślenia, że w rozpatrywanym przypadku tylko stacja PS1 uwierzytelnia się wobec stacji PS2 (w razie konieczności możliwe jest także uruchomienie dwustronnego uwierzytelniania).



Rys. 3. Kanały transmisyjne

Kanałem sterującym przesyłane są komunikaty sterujące procesem zabezpieczania transmisji. Wykorzystywane wartości zmiennej sterującej oraz oznaczenie trybów pracy systemu zabezpieczającego przedstawiono w tabeli 1. Więcej szczegółów dotyczących trybów pracy można znaleźć w [12].

Proces zabezpieczenia transmisji składa się z czterech podprocesów:

— uwierzytelniania (p. 3.1);
— transmisji kluczy publicznych asymetrycznych (p. 3.2);
— transmisji klucza symetrycznego (p. 3.3);
— transmisji danych (p. 3.4).

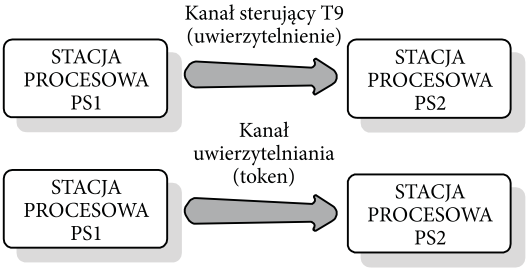
Tabela 1

Wykorzystywane tryby transmisji

Tryb	Wartość zmiennej sterującej (BIN)	Opis trybu transmisji	Wykorzystywany kanał
T1	0000000000000001	Transmisja standardowa bez zabezpieczeń	Kanał danych
T2	0000000000000010	Transmisja szyfrowana symetrycznie	Kanał danych/ transmisji kluczy
T3	0000000000000100	Transmisja szyfrowana asymetrycznie	Kanał danych
T5	0000000000001000	Transmisja klucza asymetrycznego publicznego	Kanał transmisji kluczy
T9	0000000000010000	Transmisja danych uwierzytelniających z dołączoną wartością funkcji skrótu	Kanał uwierzytelniania
T13	0001000000000000	Transmisja danych z dołączoną wartością funkcji skrótu i szyfrowaną symetrycznie	Kanał danych
T14	001000000000000	Transmisja danych z dołączoną wartością funkcji skrótu i szyfrowaną asymetrycznie	Kanał danych/ transmisji kluczy
T15	010000000000000	Transmisja danych z dołączoną wartością funkcji skrótu	Kanał danych

3.1. Podproces uwierzytelniania

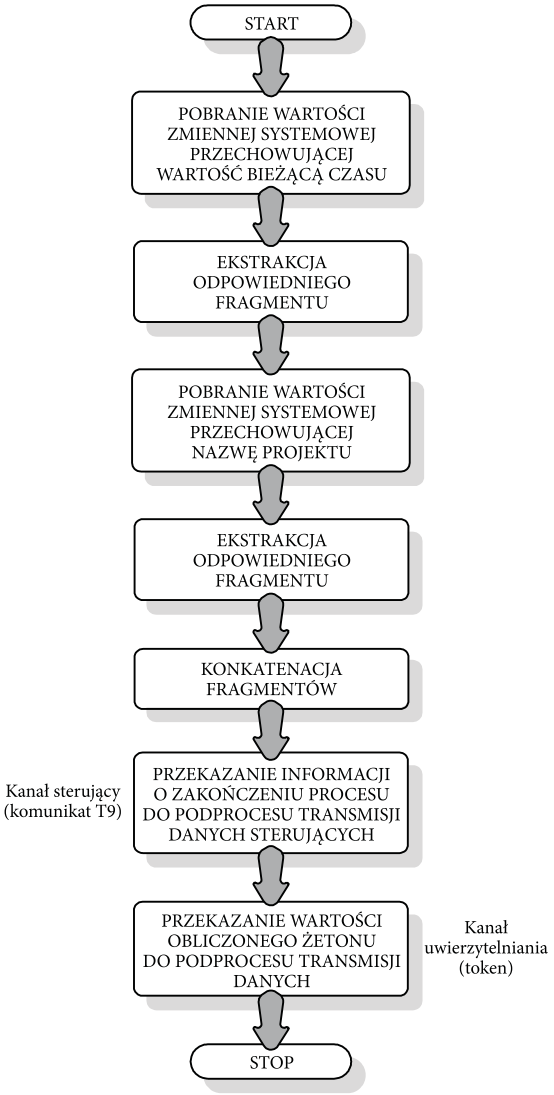
Stacja procesowa PS1 za pomocą kanału sterującego informuje (wysyła komunikat) o wysłaniu kanałem uwierzytelniania żetonu (*tokena*) programowego, który jest obliczoną wartością funkcji skrótu wg algorytmu MD5 [13] (rys. 4).



Rys. 4. Uwierzytelnianie

Żeton jest obliczany na podstawie fragmentu nazwy projektu oraz zmiennego parametru czasowego (rys. 5). Po przesłaniu następuje komparacja wartości otrzymanej od stacji PS1 oraz obliczonej przez stację PS2. Proces kończy się:

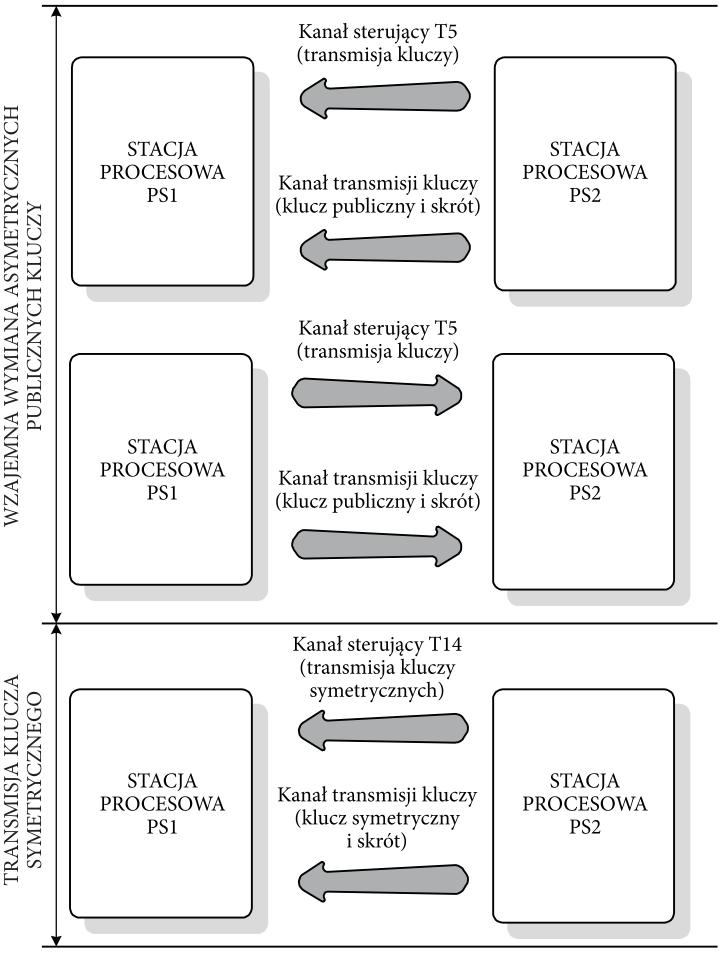
a) przekazaniem sterowania do procesu transmisji kluczy publicznych (w przypadku sukcesu komparacji);
b) powrotem do stanu oczekiwania na uwierzytelnienie (nieudane uwierzytelnienie).



Rys. 5. Algorytm procesu formowania żetonu

3.2. Podproces transmisji kluczy publicznych

W przypadku zgodności wartości tokenu odebranej od stacji PS1 i obliczonej przez stację PS2 następuje proces generowania par kluczy asymetrycznych w obydwu stacjach. Przesyłany jest klucz publiczny stacji PS2 z jednoczesnym wystawieniem odpowiedniego komunikatu sterującego (rys. 6).



Rys. 6. Podprocesy transmisji kluczy

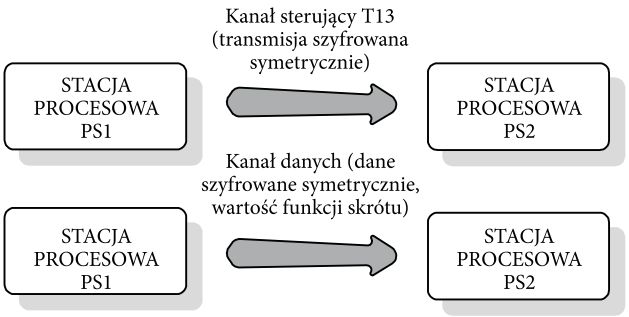
Podobnie postępuje stacja PS1, wysyłając swój klucz publiczny. W przeciwnym przypadku, tj. braku poprawnego uwierzytelnienia się stacji PS1, kolejne kroki są ignorowane, stacja PS2 przechodzi do oczekiwania, o czym wspomniano w p. 3.1.

3.3. Podproces transmisji klucza symetrycznego

W chwili pozytywnego zakończenia procesu wymiany kluczy każda ze stacji ma możliwość wysłania informacji szyfrowanej asymetrycznie, korzystając z klucza drugiej strony. Tak skonstruowany podproces transmisji klucza symetrycznego eliminuje trudność polegającą na braku tajnego sposobu przekazania drugiej stronie klucza symetrycznego. Tylko stacja udostępniająca swój klucz publiczny ma drugi z pary — prywatny — pozwalający na poprawną deszyfrację wiadomości. Otrzymany klucz publiczny umożliwia rozpoczęcie generowania przez stację PS2 klucza symetrycznego oraz wysłanie go (także przez PS2) do PS1. Klucz ten używany jest w dalszej części procesu zabezpieczonej komunikacji pomiędzy stacjami.

3.4. Podproces transmisji danych

Pozytywnie zakończony proces transmisji klucza symetrycznego daje możliwość rozpoczęcia transmisji szyfrowanej symetrycznie. Rozpoczyna się transmisja danych (na rysunku 7 z PS1 do PS2) zaszyfrowanych kluczem prywatnym, znanym tylko stacjom. Każdorazowo przed wysłaniem wartość zmiennej procesowej poddawana jest procesowi szyfrowania symetrycznego.



Rys. 7. Transmisja zabezpieczonych danych

Oprócz zaszyfrowanej wartości zmiennych procesowych do transmitowanych danych dołączana jest także wartość funkcji skrótu. Na rysunkach 5-7 oznaczono symbolicznie numery trybów pracy systemu zabezpieczającego zgodnie z tabelą 1.

4. Implementacja zabezpieczeń w sterowniku przemysłowym

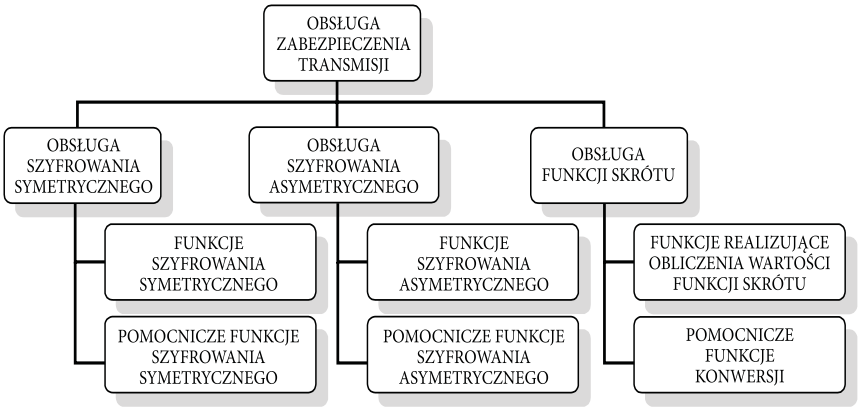
Przedstawiony w punkcie 3 proces zabezpieczania transmisji został zrealizowany w sterowniku przemysłowym oraz w jego emulatorze. Wykorzystano do tego celu języki programowania zgodne z normą PN-EN 61131 [14]:

— język bloków funkcyjnych (ang. *FBD — Function Block Diagram*);
— język ST (ang. *Structural Text*).

Funkcje zostały zaimplementowane w formie bloków funkcyjnych użytkownika umożliwiających zawarcie w strukturze bloku fragmentów mieszanego kodu w języku ST i języku blokowym FBD).

Główne funkcje systemu zabezpieczającego zostały pogrupowane i przyporządkowane do sześciu kategorii (posegregowano je tematycznie w tabelach 2-7).

Dostępne kategorie funkcji (rys. 8) to [15]:
— funkcje szyfrowania asymetrycznego (tab. 2);
— pomocnicze funkcje szyfrowania asymetrycznego (tab. 3);
— funkcje szyfrowania symetrycznego (tab. 4);
— pomocnicze funkcje szyfrowania symetrycznego (tab. 5);
— funkcje realizacji operacji obliczania wartości funkcji skrótu (tab. 6);
— pomocnicze funkcje konwersji (tab. 7).



Rys. 8. Podział funkcji realizujących zabezpieczanie transmisji

Dzięki wykorzystaniu metody tworzenia bloków użytkownika skonstruowany szablon funkcji można wykorzystać wielokrotnie.

Tabela 2

Funkcje szyfrowania asymetrycznego

Opis funkcji	Typ parametru wejściowego	Typ parametru wyjściowego
Deszyfrowanie ciągu znaków wg algorytmu RSA	łańcuch 32-znakowy (STR32), łańcuch 256-znakowy (STR256)	łańcuch 256-znakowy (STR256)
Szyfrowanie ciągu znaków wg algorytmu RSA	dwie zmienne — łańcuch 32 (STR32)	łańcuch 256-znakowy (STR256)
Generowanie klucza publicznego i prywatnego	zmienna typu data/czas (DT)	dwie zmienne — łańcuch 32-znakowy (STR32)

Tabela 3

Funkcje pomocnicze implementowane do szyfrowania asymetrycznego

Opis funkcji	Typ parametru wejściowego	Typ parametru wyjściowego
Zwracająca znak kodu ASCII	zmienna typu całkowitego bez znaku (DINT)	łańcuch 8-znakowy (STR8)
Potęgowanie i dzielenie z resztą (modulo)	trzy zmienne typu całkowitego bez znaku (DINT)	zmienna typu całkowitego bez znaku (DINT)
Obliczanie najmniejszego wspólnego dzielnika	dwie zmienne typu całkowitego bez znaku (DINT)	zmienna typu całkowitego bez znaku (DINT)
Obliczanie potęgi liczby	dwie zmienne typu całkowitego (INT)	zmienna typu całkowitego bez znaku (DINT)

Tabela 4

Funkcje szyfrowania symetrycznego

Opis funkcji	Typ parametru wejściowego	Typ parametru wyjściowego
Funkcja F algorytmu DES	łańcuch 256-znakowy (STR256), łańcuch 64-znakowy (STR64)	łańcuch 256-znakowy (STR256)
Szyfrowanie/deszyfrowanie danych wg algorytmu DES	dwie zmienne — łańcuch 64-znakowy (STR64), łańcuch 8-znakowy (STR8)	łańcuch 64-znakowy (STR64)
Obliczanie różnicy symetrycznej XOR	dwie zmienne — łańcuch 64-znakowy (STR64)	łańcuch 64-znakowy (STR64)

Tabela 5

Funkcje pomocnicze implementowane do szyfrowania symetrycznego

Opis	Typ parametru wejściowego	Typ parametru wyjściowego
Konwersja postaci binarnej ciągu znaków na tekst jawny	łańcuch 64-znakowy (STR64)	łańcuch 64-znakowy (STR64)
Konwersja postaci binarnej ciągu znaków na szesnastkową	łańcuch 64-znakowy (STR64)	łańcuch 64-znakowy (STR64)
Zamiana wartości dziesiętnej na postać binarną	zmienna typu całkowitego (INT)	łańcuch 64-znakowy (STR64)
Przesunięcie ciągu znaków	łańcuch 64-znakowy (STR64), zmienna typu całkowitego (INT)	łańcuch 64-znakowy (STR64)

Tabela 6

Funkcje implementowane do obliczania wartości funkcji skrótu

Opis funkcji	Typ parametru wejściowego	Typ parametru wyjściowego
Obliczenie skrótu wg alg. MD5	łańcuch 256-znakowy (STR256)	łańcuch 256-znakowy (STR256)
Funkcja GG (algorytm MD5)	łańcuch 8-znakowy (STR8) — siedem zmiennych	łańcuch 8-znakowy (STR8)
Funkcja II (algorytm MD5)	łańcuch 8-znakowy (STR8) — siedem zmiennych	łańcuch 8-znakowy (STR8)
Funkcja HH (algorytm MD5)	łańcuch 8-znakowy (STR8) — siedem zmiennych	łańcuch 8-znakowy (STR8)
Funkcja FF (algorytm MD5)	łańcuch 8-znakowy (STR8) — siedem zmiennych	łańcuch 8-znakowy (STR8)
Funkcja i (algorytm MD5)	łańcuch 8-znakowy (STR8) — trzy zmienne	zmienna typu całkowitego bez znaku (UDINT)
Funkcja h (algorytm MD5)	łańcuch 8-znakowy (STR8) — trzy zmienne	zmienna typu całkowitego bez znaku (UDINT)
Funkcja g (algorytm MD5)	łańcuch 8-znakowy (STR8) — trzy zmienne	zmienna typu całkowitego bez znaku (UDINT)
Funkcja f (algorytm MD5)	łańcuch 8-znakowy (STR8) — trzy zmienne	zmienna typu całkowitego bez znaku (UDINT)

Tabela 7

Pomocnicze funkcje konwersji

Opis funkcji	Typ parametru wejściowego	Typ parametru wyjściowego
Przesunięcie bitowe w lewo	podwójne słowo (DWORD), zmienna typu całkowitego (INT)	podwójne słowo (DWORD)
Zamiana pojedynczego znaku na liczbową reprezentację (ASCII)	łańcuch 8-znakowy (STR8)	output (UDINT)
Konwersja dowolnego ciągu znaków na postać szesnastkową	łańcuch 256-znakowy (STR256)	łańcuch 256-znakowy (STR256)
Konwersja ciągu znaków z postaci szesnastkowej na binarną	łańcuch 256-znakowy (STR256)	łańcuch 256-znakowy (STR256)
Konwersja postaci dziesiętnej liczby na binarną	zmienna typu całkowitego bez znaku (UDINT)	łańcuch 256-znakowy (STR256)
Uzupełnienie ciągu znaków zerami (parametr) z lewej strony	łańcuch 256-znakowy (STR256) zmienna typu całkowitego (INT)	łańcuch 256-znakowy (STR256)
Konwersja postaci binarnej liczby na dziesiętną	łańcuch 32-znakowy (STR32)	zm. typu całkowit. bez znaku (UDINT)

cd. tabeli 7

Konwersja ciągu znaków z postaci szesnastkowej na dziesiętną	łańcuch 8-znakowy (STR8)	zm. typu całkowit. bez znaku (UDINT)
Zamiana postaci dziesiętnej na szesnastkową	zmienna typu całkowitego bez znaku (UDINT)	łańcuch 8-znakowy (STR8)

5. Podsumowanie

W opracowaniu przedstawiono rozwiązanie pozwalające na zabezpieczenie przesyłanych danych pomiędzy stacjami procesowymi przemysłowego systemu sterowania. Zastosowano tu autorski algorytm uwierzytelniania stacji procesowej. Użyto algorytmu szyfrowania asymetrycznego m.in. do przesyłu tajnych kluczy symetrycznych w przemysłowym systemie sterowania, a po pozytywnym przesłaniu klucza symetrycznego — wykorzystano szyfrowanie symetryczne do przesyłu wartości zmiennych procesowych. Całością procesu zabezpieczania zarządza program sterujący wykorzystujący kanał sterujący do informowania drugiej strony o etapie procesu zabezpieczania. Za pomocą wartości przesyłanych ww. kanałem stacje mogą przełączać się pomiędzy wieloma trybami pracy elementów bezpieczeństwa. Pozwala to na użycie elementów (podprocesów) w konfiguracji preferowanej przez użytkownika.

O uniwersalności przedstawionego rozwiązania świadczą:

— implementacja funkcji i podprogramów w postaci bloków funkcyjnych użytkownika, umożliwiających ich wielokrotne wywołanie w programie sterowania;
— możliwość zastosowania opisanych elementów bezpieczeństwa do innych rodzajów komunikacji w systemie przemysłowym.

Uniezależnienie się od firmowych, zamkniętych rozwiązań osiągnięto poprzez:

— wykonanie zabezpieczeń umożliwiających zastosowanie standardowych bloków transmisji wg protokołu TCP/IP;
— eliminację stosowania dodatkowych urządzeń lub aplikacji pośredniczących tworzących szyfrowane połączenia.

Artykuł wpłynął do redakcji 29.01.2015 r. Zweryfikowaną wersję po recenzjach otrzymano 31.07.2015 r.

LITERATURA

[1] Bednarek M., *Wizualizacja procesów — laboratorium*, Oficyna Wydawnicza Politechniki Rzeszowskiej, Rzeszów, 2004.

[2] Stamp M., *Information security*, J. Wiley & Sons, Hoboken, 2006.

[3] Bednarek M., Dąbrowski T., Wiśnios M., *Diagnozowanie bezpieczeństwa komunikacji w przemysłowym systemie sterowania*, X Szkoła-konferencja „Metrologia wspomagana komputerowo", Waplewo, 27-30 maja 2014.

[4] Bednarek M., Dąbrowski T., Wiśnios M., *Diagnozowanie zagrożeń komunikacji w przemysłowym systemie sterowania*, Przegląd Elektrotechniczny, 8, 2014, 138-143.

[5] Kwiecień A., *Analiza przepływu informacji w komputerowych sieciach przemysłowych*, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, 2001.

[6] Fall K., Stevens R., *TCP/IP od środka. Protokoły*, Helion, 2013.

[7] Bednarek M., Dąbrowski T., *Koncepcja bezpiecznej transmisji danych w mobilnym systemie rozproszonym*, Materiały Międzynarodowej Konferencji „Transport XXI wieku", Ryn, 16-19.09.2013.

[8] Bednarek M., Dąbrowski T., *Koncepcja bezpiecznej transmisji danych w mobilnym systemie rozproszonym*, Prace Naukowe Politechniki Warszawskiej, seria Transport, zeszyt 96, 2013, 69-76.

[9] *Modbus Messaging On TCP/IP Implementation Guide*, October 24, 2006; http://www.Modbus-IDA.org.

[10] Iwanitz F., Lange J., *OPC Fundamentals, Implementation and Application*, Laxmi Publications Pvt Limited, 2010.

[11] Bednarek M., Dąbrowski T., Wiśnios M., *Koncepcja komunikacji bezpiecznej w systemie rozproszonym*, VIII Krajowa Konferencja DIAG 2013, Ustroń, 3-7.06.2013.

[12] Bednarek M., Dąbrowski T., Wiśnios M., *Elementy koncepcji zabezpieczenia transmisji pomiędzy stacjami diagnostycznymi*, X Szkoła-konferencja „Metrologia wspomagana komputerowo", Waplewo, 27-30 maja 2014.

[13] Stinson D.R., *Kryptografia. W teorii i praktyce*, WNT, Warszawa, 2005.

[14] Norma PN-EN 61131 Sterowniki programowalne, Część 3. Języki programowania.

[15] Bednarek M., Dąbrowski T., Marczydło D., *Elementy bezpieczeństwa przesyłu danych w przemysłowym systemie sterowania*, XLIII Zimowa Szkoła Niezawodności, Szczyrk 11-17.01.2015, s. 15.

M. BEDNAREK, T. DĄBROWSKI

Security of the data transmission in the industrial control system

Abstract. The theme of this paper is to present the data transmission security system between the stations of the industrial control system. The possible options for secure communications between process stations, as well as between process and operator station are described. Transmission security mechanism is based on algorithms for symmetric and asymmetric encryption. The authentication process uses a software token algorithm and a one-way hash function. The algorithm for establishing a secured connection between the stations, including the authentication process and encryption of data transmission is given. The process of securing the transmission consists of 4 sub-processes: (I) authentication; (II) asymmetric, public keys transmission; (III) symmetric key transmission; (IV) data transmission. The presented process of securing the transmission was realized in the industrial controller and emulator. For this purpose, programming languages in accordance with EN 61131 were used. The functions were implemented as user function blocks. This allows us to include a mixed code in the structure of the block (both: ST and FBD). Available function categories: support of the asymmetric encryption; asymmetric encryption utility functions; support of the symmetric encryption; symmetric encryption utility functions; support of the hash value calculations; utility functions of conversion.

Keywords: transmission security, encryption, authentication, industrial control system
DOI: 10.5604/12345865.1186229