

WOJSKOWA AKADEMIA TECHNICZNA

im. Jarosława Dąbrowskiego

Wydział Cybernetyki



PRACA DYPLOMOWA

STACJONARNE STUDIA MAGISTERSKIE

**Projekt i realizacja podsystemu uwierzytelniania
zdalnych użytkowników systemu informatycznego
wspomagającego funkcjonowanie
instytucji edukacyjnej**

Autor
Marcin Tomasz Wiącek

Kierownik pracy
dr inż. Wiesław Barcikowski

Warszawa 2007

Praca została obroniona 20 czerwca 2007
Niniejsza wersja elektroniczna z www.mwiacek.com zawiera późniejsze poprawki i uzupełnienia
Wykorzystanie w celach komercyjnych zabronione bez zgody autora

Zadanie do realizacji w ramach pracy dyplomowej

I. Temat pracy:

Projekt i realizacja podsystemu uwierzytelniania zdalnych użytkowników systemu informatycznego wspomagającego funkcjonowanie instytucji edukacyjnej

II. Treść zadania:

1. Analiza potrzeb instytucji edukacyjnych w zakresie uwierzytelniania zdalnych użytkowników systemów wspomagających realizację zadań statutowych
2. Charakterystyka metod i technik uwierzytelniania wykorzystywanych we współczesnych systemach informatycznych
3. Opracowanie koncepcji zaawansowanego uwierzytelniania zdalnych użytkowników systemu informatycznego wspomagającego funkcjonowanie wydziału akademickiego
4. Projekt podsystemu uwierzytelniania zdalnych użytkowników systemu informatycznego wspomagającego funkcjonowanie wydziału akademickiego
5. Implementacja wybranych elementów podsystemu uwierzytelniania zdalnych użytkowników systemu informatycznego wspomagającego funkcjonowanie wydziału akademickiego

W rezultacie wykonania pracy należy dodatkowo przedstawić planszę ilustrującą wykonaną pracę dyplomową w formie określonej według ustalonych przez kierownika pracy wymagań.

III. Terminy:

Zadanie zostało zaakceptowane przez Dziekana Wydziału Cybernetyki Wojskowej Akademii Technicznej dr hab. inż. Andrzeja Najgebauera **25 stycznia 2007** i wydane do realizacji autorowi **26 stycznia 2007**. Termin zdania ukończonej pracy **14 maja 2007**.

Spis treści

Wstęp.....	5
1. Analiza potrzeb instytucji edukacyjnych w zakresie uwierzytelniania zdalnych użytkowników systemów informatycznych.....	6
1.1. Grupy użytkowników.....	6
1.2. Wybrane potrzeby i oczekiwania grup użytkowników.....	7
1.3. Wybrane wykorzystywane systemy informatyczne i stosowanie w nich ograniczeń.....	10
1.4. Uwarunkowania prawne.....	12
1.5. Kryteria wyboru metody uwierzytelniania.....	16
2. Charakterystyka metod i technik uwierzytelniania wykorzystywanych we współczesnych systemach informatycznych.....	17
2.1. Funkcje matematyczne wykorzystywane przy uwierzytelnianiu.....	18
2.1.1. Funkcje „skrótów”.....	18
2.1.2. Szyfrowanie symetryczne.....	20
2.1.3. Szyfrowanie asymetryczne.....	22
2.2. Schematy przeprowadzania procesu uwierzytelniania w systemach informatycznych.....	24
2.2.1. Porównywanie z danymi wzorcowymi.....	24
2.2.2. „Wyzwanie - odpowiedź”.....	26
2.2.3. Uwierzytelnianie za pomocą kluczy publicznych.....	28
2.3. Metody wprowadzania danych identyfikacyjnych do systemów wykorzystujące standardowe urządzenia komputerowe.....	32
2.3.1. Wprowadzanie z użyciem myszki/rysika albo klawiatury.....	32
2.3.1.1. Hasła wielokrotne.....	34
2.3.1.2. Odpowiedzi na pytania.....	40
2.3.1.3. Captcha i Hip.....	40
2.3.2. Wprowadzanie z plików.....	42
2.4. Metody wprowadzania danych identyfikacyjnych do systemów wymagające posiadania specjalnych przedmiotów.....	43
2.4.1. Numery seryjne programów.....	44
2.4.2. Hasła jednorazowe.....	45
2.4.3. Karty stykowe z paskiem magnetycznym.....	46
2.4.4. Karty stykowe z układami elektronicznymi.....	47
2.4.5. Karty i przedmioty zbliżeniowe.....	50
2.4.6. Tokeny.....	51
2.4.7. TPM.....	52
2.4.8. Klucze sprzętowe.....	53
2.5. Metody wprowadzania danych identyfikacyjnych do systemów związane z cechami użytkowników.....	53
2.5.1. Metody biometryczne.....	53
2.5.1.1. Odciski linii papilarnych.....	55
2.5.1.2. Rozpoznawanie kształtu twarzy.....	57
2.5.1.3. Badanie tęczy oka.....	58
2.5.1.4. Rozpoznawanie wzoru żył.....	59
2.5.1.5. Badanie kształtu dłoni.....	60
2.5.1.6. Badanie podpisu odręcznego.....	60
2.5.1.7. Rozpoznawanie głosu.....	60
2.5.1.8. Badanie siatkówki oka.....	61

2.5.1.9. Identyfikacja związana z uchem.....	61
2.5.2. Uwierzytelnianie przez innego człowieka.....	62
2.6. Metody nie wymagające wprowadzania danych przez użytkownika.....	62
3. Podsystem uwierzytelniania zdalnych użytkowników systemu informatycznego do wykorzystania w wydziale akademickim.....	64
3.1. Koncepcja.....	64
3.1.1. Wybór sposobu wprowadzania danych identyfikacyjnych przez użytkownika.....	64
3.1.2. Schemat uwierzytelniania.....	65
3.1.3. Wady i zalety.....	67
3.2. Projekt.....	68
3.2.1. Studium wykonalności.....	69
3.2.2. Specyfikacja wymagań.....	69
3.2.2.1. Wymagania funkcjonalne.....	70
3.2.2.2. Wymagania pozafunkcjonalne.....	70
3.2.3. Architektura podsystemu.....	71
3.2.3.1. Struktura bazy danych.....	72
3.2.3.2. Moduł uwierzytelniania.....	74
3.2.3.3. Moduły administratora.....	76
3.3. Implementacja.....	77
3.3.1. Wybór technologii.....	77
3.3.2. Realizacja podsystemu.....	78
3.3.3. Testowanie podsystemu.....	79
Podsumowanie.....	85
Bibliografia.....	87
Indeks.....	94
Spis rysunków.....	95
Załącznik A	97

Wstęp

Układy scalone i technika mikrokomputerowa na dobre zagościły w naszym codziennym życiu. Technologie z tym związane pozwoliły na uzyskanie wielu nowych możliwości, jednakże jesteśmy również coraz bardziej od nich uzależnieni. Komputerom powierzamy coraz częściej nasze pieniądze i różnego rodzaju sekrety. To, co jest zapisane w ich pamięci, zaczyna być nawet dowodem w sprawach karnych, skarbowych, itp. Coraz więcej mówi się wręcz o pełnym kontrolowaniu z ich wykorzystaniem tego, co i jak robimy.

Celem pracy jest przygotowanie podsystemu, który można by wykorzystać w systemach informatycznych w instytucji edukacyjnej (wydziale akademickim) do sprawdzania tożsamości użytkowników tak, aby budziło to jak najmniejsze wątpliwości.

W tym celu zostaną zidentyfikowane te systemy informatyczne, w którym istotna jest weryfikacja tożsamości użytkownika. Podane zostaną również skutki nieprawidłowego jej określenia (m.in. przedstawione zostaną odpowiednie elementy prawa obowiązującego w Polsce) po to, aby stwierdzić, na ile tematyka ta jest tam ważna i jakie dane szczególnie powinny być chronione. Na tej podstawie zostaną wysnute pewne wnioski dotyczące kryteriów, według których należy wybrać metodę uwierzytelniania.

Następnie zostaną pokazane wynalezione do tej chwili grupy metod pozwalających na sprawdzanie tożsamości w systemach informatycznych (ponieważ rozważane będą tutaj nie tylko instytucje edukacyjne, ta część będzie zdecydowanie największa). Położony zostanie nacisk raczej na ogólne przedstawienie słabości opisywanych rozwiązań niż szczegółowe opisywanie dostępnych na rynku ich implementacji. Będzie to zrobione bez wdawania się w różne detale techniczne tak, aby było zrozumiałe dla możliwie szerokiego grona ludzi.

Na końcu zostanie przygotowany wspomniany już podsystem (w formie projektu). W założeniu ma on pozwolić na przeprowadzenie czynności uwierzytelniania w sposób pozwalający uniknąć opisanych wcześniej słabości i ma być łatwo integrowalny z obecnymi już w zaprezentowanych organizacjach rozwiązaniami. Najważniejsze będzie tutaj szczegółowe przedstawienie pewnej przyjętej koncepcji tak, aby można było ją prosto zakodować w różnych językach programowania i narzędziach (zaproponowana implementacja będzie tutaj raczej jedynie dodatkiem do niej).

Należy zauważyć, że różne przedstawione w tej pracy rozwiązania mogą szybko nabrać znaczenia jedynie historycznego (choćby z uwagi na dynamikę postępu technologicznego), niemniej jednak przynajmniej pewne jej elementy powinny pozostać długo prawdziwe i będą mogły służyć jako podstawa do przeprowadzenia jeszcze głębszych i dokładniejszych rozważań dotyczących opisywanych kwestii.

1. Analiza potrzeb instytucji edukacyjnych w zakresie uwierzytelniania zdalnych użytkowników systemów informatycznych

Podstawowym celem każdej instytucji edukacyjnej jest pomoc określonym grupom ludzi w uzyskaniu przez nich określonej wiedzy. Przykładowo w Wojskowej Akademii Technicznej (w dalszych fragmentach pojawiać się będzie dużo przykładów związanych z uczelniami wyższymi i wydziałami akademickim ze względu na przydatność do rozważań o projekcie w rozdziale 3, natomiast wspomniana placówka pojawiać się będzie najczęściej m.in. z uwagi na bliskość autorowi tej pracy dyplomowej) zostało to zdefiniowane następująco:

„§ 5.¹

2. Do podstawowych zadań Akademii należy:

- 1) kształcenie studentów w celu ich przygotowania do pracy zawodowej,
- 2) kształcenie kadr specjalistycznych i dowódczych dla jednostek wojskowych oraz innych jednostek organizacyjnych podległych lub nadzorowanych przez Ministra Obrony Narodowej,
- 3) kształcenie kadr dydaktycznych i naukowych dla szkolnictwa wojskowego i jednostek badawczo-rozwojowych nadzorowanych przez Ministra Obrony Narodowej,
- 4) doskonalenie zawodowe osób dla potrzeb bezpieczeństwa państwa,
- 5) kształcenie i szkolenie cudzoziemców zgodnie z zadaniami postawionymi przez Ministra Obrony Narodowej,
- 7) realizacja form kształcenia ustawicznego”

Aby to osiągnąć, konieczne jest stosowanie różnych środków. Poniżej zostanie omówione, kto i jak w tego typu instytucjach może wykorzystać do tego systemy informatyczne. Będzie to punktem wyjściowym do przeprowadzenia rozważań na temat uwierzytelniania w tych systemach.

1.1. Grupy użytkowników

Jak wspomniano wcześniej, celem każdej instytucji jest pomoc określonym ludziom w uzyskaniu przez nich wiedzy. Z tego powodu podstawową grupą będą osoby uczące się. Bez nich instytucja nie ma szans istnieć – nie mogłaby realizować swojej podstawowej roli usługowej, są oni również źródłem jej dochodu (płacą samodzielnie albo ich naukę finansują inne instytucje takie jak Ministerstwo Obrony Narodowej albo Ministerstwo Edukacji Narodowej). Można oczywiście stwierdzić, że bez nich dana placówka również mogłaby istnieć (utrzymywałaby się np. z usług świadczonych na rzecz firm) - wtedy jednak byłaby to już zwykła firma, a nie instytucja edukacyjna. Z wyżej wymienionych powodów na rysunku obok osoby uczące zostały oznaczone kolorem czerwonym. W zależności od rodzaju placówki nazywa się ich różnie – w szkole, gimnazjum, liceum są to uczniowie (ewentualnie abiturienti), w uczelni studenci czy też słuchacze.

¹ Statut Wojskowej Akademii Technicznej im. Jarosława Dąbrowskiego w Warszawie (załącznik do uchwały Senatu WAT Nr 40/II/2006 z dnia 23 lutego 2006 wraz ze zmianami wniesionymi uchwałą Senatu WAT nr 71/II/2006 z dnia 12 października 2006) dostępny pod adresem <http://www.wat.edu.pl/0001/statut.pdf>

Drugą bardzo ważną grupą jest kadra dydaktyczna – grupa ludzi, która przekazuje wiedzę. Na rysunku pokazano ich jako będących na pograniczu instytucji – zdarza się, że dana jednostka jedynie „wynajmuje” na jakiś czas specjalistów z danej dziedziny (przykład: mogą być to osoby prowadzące jednostkowe wykłady dla słuchaczy). Oczywiście, im więcej znanych i uznanych w swoich środowiskach ludzi pracuje na stałe na rzecz instytucji, tym większe ma ona znaczenie oraz renomę i jest chętniej wybierana przez różne osoby jako miejsce przyszłej edukacji.



Rysunek 1. Grupy użytkowników związane z instytucją edukacyjną (źródło: opracowanie własne)

Mamy również pracowników zapewniających sprawność dostępnej infrastruktury (będą to np. elektrycy, hydraulicy lub osoby sprzątające) i różne usługi pomocniczne (np. ochrona czy obsługa wszelkiego rodzaju barów i ośrodków żywieniowych działających na terenie jednostki edukacyjnej). Osoby te są często (nie jest to reguła) zatrudniane formalnie przez firmy zewnętrzne (które mają podpisane odpowiednie umowy z jednostką edukacyjną).

Aby całość mogła funkcjonować, konieczne jest istnienie kadry zarządzającej. Nadzoruje ona działalność wymienionych wcześniej grup, podejmuje decyzje dotyczące ich składów, jak również stanowi w sposób ostateczny o przyszłości danej placówki (np. jej ofercie edukacyjnej czy sposobie rozdysponowania finansów). W przypadku uczelni wyższej jest to najczęściej rektor, zastępca, prorektorzy, dziekani, prodziekani, a na niższym szczeblu różni pracownicy dziekanatów.

W przeprowadzanych dalej rozważaniach pewne znaczenie będą mieli również kandydaci (ludzie, którzy chcą się uczyć w danej instytucji i poczynili w tym kierunku pewne określone kroki takie jak np. zarejestrowanie się).

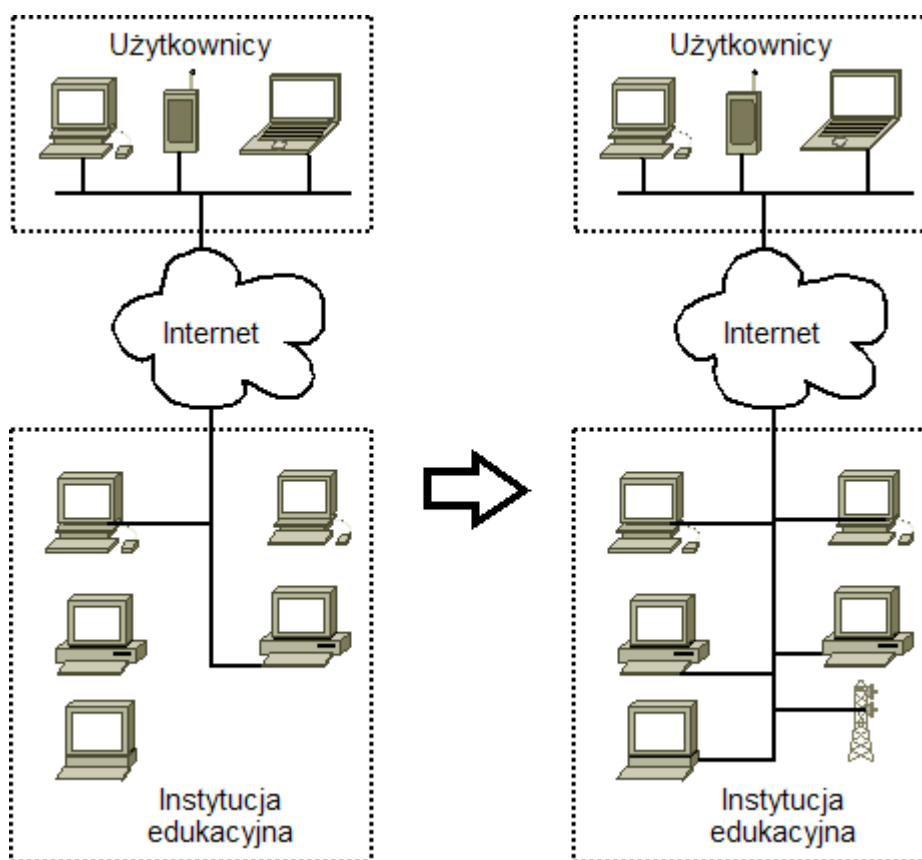
Warto jeszcze wspomnieć o niektórych osobach spoza instytucji – tych, którzy mogą być z nią związani chwilowo (np. przeglądając jej stronę www lub będąc w jej siedzibie).

1.2. Wybrane potrzeby i oczekiwania grup użytkowników

Jak można zauważyć, szeroko rozumiana technika komputerowa wypiera w wielu dziedzinach rozwiązania tradycyjne. Daje ona bowiem szereg korzyści – umożliwia zmniejszenie rozmiarów i wagi urządzeń przy zwiększeniu ich możliwości czy też pozwala na obniżkę różnego rodzaju kosztów. Coraz więcej ludzi (warto przypomnieć – mogą oni w szczególności należeć do przedstawionych wcześniej

grup) ma dostęp do komputera oraz Internetu nawet w domu...i chciałoby z ich użyciem załatwiać różne sprawy czy też otrzymać pewne informacje.

Wszyscy prawdopodobnie będą potrzebować dostępu do podstawowych danych o instytucji takich jak adres czy numery telefonów, mogą być również zainteresowani historią jednostki edukacyjnej czy też danymi o różnych wydarzeniach, które są tam organizowane (w przypadku Wojskowej Akademii Technicznej takim wydarzeniem jest np. Dzień Podchorążego, w trakcie którego dokonuje się inscenizacji wydarzeń z Powstania Listopadowego). Dla kandydatów ważne będą informacje o naborze takie jak daty, wymagane kwoty czy dokumenty, kryteria przyjęcia, itp. Ludzie z innych grup mogliby być zainteresowani dostępem do komunikatów władz (np. studenci – do ogłoszeń dziekanatu) czy też swoich rozkładów zajęć.



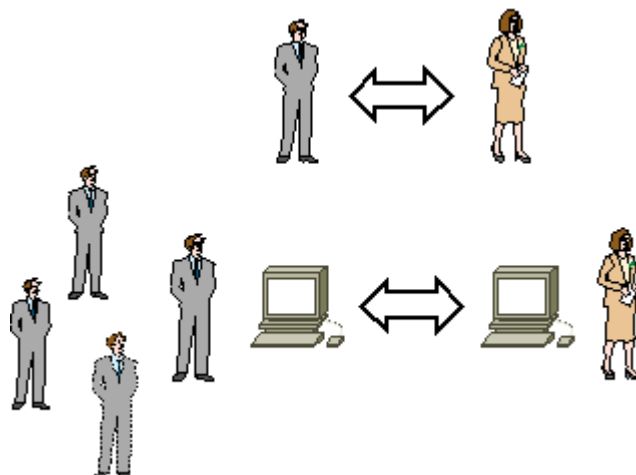
Rysunek 2. Tendencją widoczną również w instytucjach edukacyjnych jest łączenie jak największej liczby systemów ze sobą i udostępnianie ich usług bezprzewodowo oraz w Internecie (źródło: opracowanie własne)

Dla kandydatów ważna będzie:

- możliwość rejestracji i poprawiania swoich danych
- możliwość wykonania przelewu - właściwie wystarczy podać numer konta bankowego instytucji (i po stronie instytucji sprawdzać tylko, od kogo wypłynęły pieniądze), w bardziej zaawansowanych rozwiązaniach można zastosować usługi typu mTRANSFER², co zautomatyzuje i przyspieszy ten proces.

² <http://www.mbank.com.pl/oferta/mtransfer/index.html>

- sprawdzenie wyników napisanego przez siebie sprawdzianu. Oczywiście w chwili obecnej można przyjąć, że złożenie czy odbiór niektórych wymaganych dokumentów wymaga już osobistej obecności kandydata, podobnie system informatyczny instytucji niekoniecznie może pozwolić na napisanie przez tę osobę sprawdzianu kwalifikacyjnego (mogłoby się bowiem zdarzyć, że człowiek ten siedząc przy swoim komputerze korzystałby z pomocy osób trzecich)



Rysunek 3. W systemach informatycznych problemem jest przeprowadzenie kontroli tego, czy dane są tam wprowadzane samodzielnie przez użytkownika czy nie (źródło: opracowanie własne)

Z kolei osoby uczące się w danej instytucji mogą być zainteresowane:

- dostępem do wszystkich swoich ocen
- wszystkimi materiałami prezentowanymi im przez kadrę dydaktyczną. Te ostatnie są zresztą tworzone coraz częściej z użyciem komputerów (wszelkiego rodzaju dokumenty czy prezentacje wyświetlane na wykładach), więc wystarczy tylko umieścić odpowiednie pliki w znanym zainteresowanym osobom miejscu.
- uzyskaniem pełnego kontaktu z kadrą dydaktyczną – zadawania pytań i uzyskiwania odpowiedzi, ale również wysyłania wykonanych zadań (o ile oczywiście rozwiązanie można przedstawić w formie elektronicznej)
- możliwością odbywania wszystkich zajęć oraz zdawania sprawdzianów w tej formie (prawdopodobnie jednak nigdy to nie zostanie w pełni zrealizowane nie tylko z uwagi na problemy z kontrolą samodzielności, ale po prostu z uwagi na to, że siedzenie przy komputerze po prostu nie może zastąpić np. kontaktu ze specjalistycznym sprzętem)

Jak wspomniano wcześniej, w celu załatwienia wielu spraw należy udać się do odpowiednich miejsc w instytucji. Bardzo wygodne byłoby umożliwienie bezpośredniego wpłacania związanych z tym pieniędzy lub składania związanych z tym wniosków.

Przykładowo: dzisiaj student, który potrzebuje zaświadczenia o studiowaniu (w celu przedstawienia np. w ZUS), musi udać się w odpowiednich godzinach do dziekanatu i złożyć prośbę o wydanie, po kilku dniach musi tam pójść ponownie (żeby sprawdzić, czy wniosek jest już gotowy albo żeby go odebrać). Gdyby system informatyczny instytucji umożliwił wysłanie prośby o wydanie wniosku w

sposób pozwalający w odpowiednim stopniu na zweryfikowanie tożsamości osoby wysyłającej, a później na sprawdzenie (tej samej osobie), że dokument taki jest gotowy, zmniejszyłoby to liczbę niezbędnych wizyt w dziekanacie. Można jeszcze sobie wyobrazić, że w ogóle najlepiej byłoby, gdyby dziekanat potrafił sam wysłać taki wniosek do ZUS (byłoby to zdecydowanie najwygodniejsze - student nie byłby zmuszony do wykonania żadnych dodatkowych czynności).

Kadra nauczycielska mogłaby być zainteresowana pełniejszym dostępem do tworzonych przez siebie dokumentów. Druga ważna potrzeba to chęć ułatwienia sobie różnych czynności administracyjnych związanych z nauczaniem (takich jak przekazywanie władzom ocen osób uczących się, wprowadzanie korekt w planie zajęć, możliwość łatwego udostępniania materiałów szkoleniowych, itp.) oraz czynności związanych z prowadzonymi projektami badawczymi (np. zdalna kontrola przyrządów i pobieranie uzyskanych z nich wyników). Podobne wymagania mogą mieć również władze instytucji.

1.3. Wybrane wykorzystywane systemy informatyczne i stosowanie w nich ograniczeń

Można wskazać pewne klasy lub wręcz nazwy rozwiązań informatycznych, które pomogą spełnić m.in. część opisanych w poprzednim podrozdziale potrzeb:

- instytucje to również organizacje. W celu bardziej efektywnego zarządzania wykorzystuje się tam coraz bardziej wyspecjalizowane rozwiązania wspomagające (w dziekanatach może to być np. pakiet Sokrates³, w bibliotekach Sowa⁴, ale również całe systemy klasy np. ERP). Coraz częściej stosuje się różnego rodzaju monitoring oparty o technikę cyfrową (rejestrowany obraz jest zapisywany na dyskach twardych czy płytach) albo kontrolę dostępu (np. przy wjeździe na parking).
- zdarza się, iż dana szkoła czy też uczelnia prowadzi również badania komercyjne albo wykonuje specjalistyczne ekspertyzy. Wspomniany już statut WAT definiuje to następująco:

„§ 5.¹

2. Do podstawowych zadań Akademii należy:

9) *prowadzenie badań naukowych i prac rozwojowych, wykonywanie ekspertyz, diagnoz i prognoz oraz świadczenie usług badawczych ze szczególnym uwzględnieniem potrzeb obronności i bezpieczeństwa państwa,*”

Mogą się tutaj pojawiać wyniki z instrumentów badawczych (zapisane np. w bazach danych), projekty eksperymentów czy urządzeń (stworzone w specjalistycznych pakietach), próbki do badań (np. w postaci zeskanowanych zdjęć), itp.

- nauczanie osób (kontakt z wykładowcami, wysyłanie zadań, itp.) zwane jest inaczej „e-learningiem” albo „nauczaniem na odległość”, a pewne rozwiązania tego typu już istnieją np. w akademii Cisco (Cisco Networking Academy)⁵, Wydziale Cybernetyki Wojskowej Akademii Technicznej, Politechnice

³ <http://www.cs.put.poznan.pl/sokrates/index.html>

⁴ <http://www.sokrates.pl/>

⁵ <http://cisco.netacad.net>

Warszawskiej (tzw. OKNO - Ośrodek Kształcenia na Odległość⁶) czy też Uniwersytecie Warszawskim (tzw. COME - Centrum Otwartej i Multimedialnej Komunikacji⁷). Są to implementacje własne, pakiety klasy CMS (Course Management System) jak Moodle⁸, w tej roli częściowo sprawdzają się również różne systemy forów dyskusyjnych.

Listę tę można wydłużać w nieskończoność, ponieważ danych i systemów będzie coraz więcej (choćby dlatego, że w instytucjach wprowadza się cały czas nowe usługi). Wymusza to po prostu konkurencja. Przykładowo: jeżeli Uniwersytet Jagielloński udostępnia możliwość rejestracji kandydatów poprzez Internet⁹, brak tej funkcjonalności w innych uczelniach może wywoływać wrażenie u potencjalnych kandydatów, iż są one mniej „nowoczesne” albo że nie posiadają odpowiednich środków technicznych.

Pewne ogólne wnioski można wysnuć natomiast już teraz:

- coraz więcej systemów udostępnia swój interfejs użytkownikom w standardowych przeglądarkach www
- właściwie w KAŻDEJ przedstawionej wcześniej sytuacji niezbędne jest wprowadzenie jakichś ograniczeń (blokowania dokonywania zmian lub blokowania dostępu w ogóle) przynajmniej niektórym użytkownikom.

Można powiedzieć, że zabezpieczenia w systemach informatycznych pojawiają się:

- aby przynajmniej częściowo uzyskać kontrolę nad tym, kto wykonał daną czynność (np. wysłał wykładowcy określony plik)
- ponieważ były obecne w zastępowanych procedurach (przykładowo: skoro w dziekanacie pismo z prośbą o zaświadczenie mógł złożyć tylko zainteresowany, teraz też tak będzie)
- ze względów prawnych (jest to wymagane przez odpowiednie umowy podpisane przez instytucję, wynika z obowiązujących norm albo przepisów – temat ten zostanie omówiony szerzej dalej)
- z powodów czysto ekonomicznych - wiele z tworzonych materiałów (np. dla słuchaczy) można sprzedawać różnym ludziom (ci zapłacą za nie tylko wtedy, jeżeli nie będą dostępne za darmo dla wszystkich)
- gdyż niezbędne jest to do zachowania integralności i spójności danych – przykładowo, jeżeli dane o wpłatach za czesne mogliby zmieniać sami studenci, tak naprawdę nie byłyby one w ogóle wiarygodne

W części przypadków wystarczy oczywiście odłączać systemy z danymi od reszty świata i w ten sposób uzyskać wymaganą kontrolę. Rozwiązanie to sprawdzi się jednak wyłącznie w pojedynczych przypadkach i będzie niezgodne z ogólnym trendem do łączenia wszystkiego ze wszystkim. Konieczne jest zastosowanie innych środków, a dokładniej szerokie wykorzystywanie dwustopniowego procesu: uwierzytelniania i autoryzacji. Pierwsza część tego procesu polega na sprawdzeniu tożsamości użytkownika (który chce uzyskać dostęp do danych), drugi polega na

⁶ <http://www.okno.pw.edu.pl/>

⁷ <http://www.come.uw.edu.pl/>

⁸ <http://moodle.org/>

⁹ <https://www.erk.uj.edu.pl/>

sprawdzeniu, czy ma on prawo uzyskać żądany dostęp do danych (i jeżeli nie, to odrzucić go).

W dalszej części uwierzytelnianie będzie rozumiane jako zestaw czynności wykonywanych po odebraniu danych przez system informatyczny polegających na ustaleniu, czy dane zostały rzeczywiście wysłane przez przedstawiający się systemowi informatycznemu podmiot (osobę, komputer, urządzenie lub usługę). Obejmuje on sprawdzanie tożsamości użytkowników przy próbie dostępu do systemu informatycznego (np. czy osoba instalująca/używająca programy w systemie informatycznym jest ich prawdziwym użytkownikiem/właścicielem), a dokładniej polega na sprawdzeniu, czy dane w formie elektronicznej przekazywane do systemu zostały rzeczywiście przygotowane przez odpowiednie osoby.

1.4. Uwarunkowania prawne

Część z danych przechowywanych przez instytucje musi być przetwarzana w sposób określony prawnie. Najłatwiej pokazać to na przykładzie obowiązującej w Polsce ustawy o ochronie danych osobowych. Definiuje ona ochronę następujących informacji:

„Art. 6¹⁰

1. *W rozumieniu ustawy za dane osobowe uważa się wszelkie informacje dotyczące zidentyfikowanej lub możliwej do zidentyfikowania osoby fizycznej.*
2. *Osobą możliwą do zidentyfikowania jest osoba, której tożsamość można określić bezpośrednio lub pośrednio, w szczególności przez powołanie się na numer identyfikacyjny albo jeden lub kilka specyficznych czynników określających jej cechy fizyczne, fizjologiczne, umysłowe, ekonomiczne, kulturowe lub społeczne.*
3. *Informacji nie uważa się za umożliwiającą określenie tożsamości osoby, jeżeli wymagałoby to nadmiernych kosztów, czasu lub działań.”*

Nasuwa się tutaj podstawowy chyba wniosek – za dane osobowe można uznać szereg informacji, nie tylko przytaczany jako przykład przez wiele źródeł PESEL. Ustawa podaje również pewne uwagi dotyczące sposobu ochrony danych:

„Art. 36¹⁰

1. *Administrator danych jest obowiązany zastosować środki techniczne i organizacyjne zapewniające ochronę przetwarzanych danych osobowych odpowiednią do zagrożeń oraz kategorii danych objętych ochroną, a w szczególności powinien zabezpieczyć dane przed ich udostępnieniem osobom nieupoważnionym, zabranieniem przez osobę nieuprawnioną, przetwarzaniem z naruszeniem ustawy oraz zmianą, utratą, uszkodzeniem lub zniszczeniem.*
2. *Administrator danych prowadzi dokumentację opisującą sposób przetwarzania danych oraz środki, o których mowa w ust. 1.*
3. *Administrator danych wyznacza administratora bezpieczeństwa informacji, nadzorującego przestrzeganie zasad ochrony, o których mowa w ust. 1, chyba że sam wykonuje te czynności.”*

Pojawiają się także inne konkretne wymagania:

¹⁰ Ustawa z dnia 29 sierpnia 1997 o ochronie danych osobowych (tekst jednolity z Dziennika Ustaw z 2002 Nr 101, poz. 926 ze zmianami z dnia 22 stycznia 2004 z Dziennika Ustaw z 2004 Nr 33, poz. 285)

„Art. 37¹⁰

Do przetwarzania danych mogą być dopuszczone wyłącznie osoby posiadające upoważnienie nadane przez administratora danych.

Art. 38

Administrator danych jest obowiązany zapewnić kontrolę nad tym, jakie dane osobowe, kiedy i przez kogo zostały do zbioru wprowadzone oraz komu są przekazywane.

Art. 39

1. Administrator danych prowadzi ewidencję osób upoważnionych do ich przetwarzania, która powinna zawierać:

- 1) imię i nazwisko osoby upoważnionej,*
- 2) datę nadania i ustania oraz zakres upoważnienia do przetwarzania danych osobowych,*
- 3) identyfikator, jeżeli dane są przetwarzane w systemie informatycznym.”*

Niezbędne jest więc stosowanie środków kontroli nad dostępem takich jak uwierzytelnianie i autoryzacja, za zaniedbania w tej dziedzinie grożą odpowiednie kary:

„Art. 51¹⁰

- 1. Kto administrując zbiorem danych lub będąc obowiązany do ochrony danych osobowych udostępnia je lub umożliwia dostęp do nich osobom nieupoważnionym, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 2.*
- 2. Jeżeli sprawca działa nieumyślnie, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do roku.*

Art. 52

Kto administrując danymi narusza choćby nieumyślnie obowiązek zabezpieczenia ich przed zabraniem przez osobę nieuprawnioną, uszkodzeniem lub zniszczeniem, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do roku.”

Informacja o stosowaniu przez instytucję wymienionej ustawy bywa czasami wręcz umieszczana w statucie:

„§ 6.¹

- 3. Dane osobowe studenta podlegają ochronie w zakresie uregulowanym w ustawie z dnia 29 sierpnia 1997 r. o ochronie danych osobowych.”*

Należy zauważyć, iż oprócz odpowiedzialności prawnej w przypadku „wycieku” chronionych danych instytucja mogłaby ponieść duży uszczerbek na swoim wizerunku. Zaniedbania w tej dziedzinie naprawę się nie opłacają.

Warto wspomnieć również o prawie autorskim. Jeżeli instytucja zakupuje pewną liczbę programów czy też dokumentacji, musi zachowywać kontrolę nad ich dystrybucją udostępniając je zgodnie z warunkami ich licencji. W przeciwnym wypadku można zostać oskarżonym i skazanym za łamanie ustawy o prawie autorskim:

„Art. 116¹¹

- 1. Kto bez uprawnienia albo wbrew jego warunkom rozpowszechnia cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystyczne wykonanie, fonogram, wideogram lub nadanie, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 2.*
- 2. Jeżeli sprawca dopuszcza się czynu określonego w ust. 1 w celu osiągnięcia korzyści majątkowej, podlega karze pozbawienia wolności do lat 3.*
- 3. Jeżeli sprawca uczynił sobie z popełniania przestępstwa określonego w ust. 1 stałe źródło dochodu albo działalność przestępną, określoną w ust. 1, organizuje lub nią kieruje, podlega karze pozbawienia wolności od 6 miesięcy do lat 5.*
- 4. Jeżeli sprawca czynu określonego w ust. 1 działa nieumyślnie, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do roku.”*

W zależności od zaistniałej sytuacji zastosowanie mogą mieć tutaj również inne artykuły w/w ustawy. Szczególnie dotkliwe dla instytucji może być działanie odpowiednich organów ścigania związane z artykułem 121 (konkretnie mówiąc - przypadkowi może ulec m.in. sprzęt komputerowy przechowyujący dane):

„Art. 121¹¹

- 1. W wypadku skazania za czyn określony w art. 115, 116, 117, 118 lub 118, sąd orzeka przepadek przedmiotów pochodzących z przestępstwa, chociażby nie były własnością sprawcy.*
- 2. W wypadku skazania za czyn określony w art. 115, 116, 117 lub 118, sąd może orzec przepadek przedmiotów służących do popełnienia przestępstwa, chociażby nie były własnością sprawcy.”*

W chwili obecnej zdarza się nawet (m.in. właśnie ze względu na lepsze możliwości zabezpieczenia i fakt, że odpowiedzialność prawna zostaje przeniesiona na inny podmiot), iż część informacji i oprogramowania udostępnianych np. przez uczelnie swoim studentom pobierana jest z wyspecjalizowanych serwerów firm trzecich. Przykładem jest program MSDNAA (MicroSoft Developer Network Academic Alliance¹²) – Wojskowa Akademia Techniczna i inne instytucje biorące w nim udział mają swoje konto na serwerze firmy e-academy Inc. z siedzibą w Kanadzie.

Wiele danych w instytucji edukacyjnej jest niezbędnych dla jej istnienia. Okazuje się, że muszą istnieć pewne warunki przechowywania tych danych, aby w przypadku ich kradzieży polskie organy ścigania mogły zająć się poszukiwaniem i skazaniem sprawców. Definiuje to Kodeks Karny¹³, gdzie mowa jest o przełamaniu zabezpieczeń (uwierzytelnianie jest zaś takim zabezpieczeniem):

„art. 267.

§1. Kto bez uprawnienia uzyskuje informacje dla niego nie przeznaczoną, otwierając zamknięte pismo, podłączając się do przewodu służącego do przekazywania informacji lub przełamując elektroniczne, magnetyczne albo inne szczególne jej zabezpieczenie, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 2.”

¹¹ Ustawa z dnia 4 lutego 1994 o prawie autorskim i prawach pokrewnych (Dziennik Ustaw z 1994 Nr 24 poz. 83)

¹² <http://msdn.microsoft.com/academic/>

¹³ Ustawa z dnia 6 czerwca 1997 – Kodeks Karny (Dziennik Ustaw z 1997 nr 88 poz. 553)

W przypadku niektórych instytucji edukacyjnych (np. WAT) można również stwierdzić, iż obowiązuje je (np. na podstawie ich Statutu) ustawa o ochronie informacji niejawnych wraz z odpowiednimi przepisami wykonawczymi:

„§ 5.¹

2. Do podstawowych zadań Akademii należy:

10) wykonywanie zadań jednostki wojskowej, a w tym:

e) zapewnienie ochrony elementów wojskowych uczelni oraz ochrony informacji niejawnych i kancelarii tajnej,”

W ustawie tej można zaś przeczytać znów o stosowaniu zabezpieczeń i prowadzeniu ewidencji dostępu do danych:

„Art. 2.¹⁴

W rozumieniu ustawy:

5) dokumentem – jest każda utrwalona informacja niejawna, w szczególności na (...) nośnikach do zapisów informacji w postaci cyfrowej i na taśmach elektromagnetycznych, także w formie (...) dysku optycznego, (...), jak również informacja niejawna utrwalona na elektronicznych nośnikach danych;

Art.18.

Pełnomocnik ochrony kieruje wyodrębnioną, wyspecjalizowaną komórką organizacyjną do spraw ochrony informacji niejawnych, zwaną dalej „pionem ochrony”, do której zadań należy:

- a) zapewnienie ochrony informacji niejawnych, w tym ich ochrony fizycznej;
- b) zapewnienie ochrony systemów i sieci teleinformatycznych, w których są wytwarzane, przetwarzane, przechowywane lub przekazywane informacje niejawne;
- c) kontrola ochrony informacji niejawnych oraz przestrzegania przepisów o ochronie tych informacji;
- d) okresowa kontrola ewidencji, materiałów i obiegu dokumentów;”

Należy również zwrócić na jeden aspekt prawny – część instytucji (szczególnie uczelni wyższych) udostępnia na swoim terenie dostęp do Internetu (np. poprzez sieć WiFi). Powinna zostać wprowadzona tam przynajmniej w podstawowym zakresie kontrola nad tym, kto i w jakim celu będzie z tego korzystał. Dzieje się tak dlatego, ponieważ:

1. w przypadku nieuprawnionego dystrybuowania lub pobierania materiałów chronionych prawami autorskimi instytucja może zostać pociągnięta do odpowiedzialności cywilnej (gdy będzie natomiast dysponować informacjami o tym, kto wykonywał wspomniane czynności, może wskazać winnych)
2. w przypadku włamania do zabezpieczonej sieci instytucji istnieje możliwość ścigania włamywacza na podstawie ustawy o ochronie niektórych usług świadczonych drogą elektroniczną opartych lub polegających na dostępie warunkowym¹⁵

¹⁴ Ustawa z dnia 22 stycznia 1999 o ochronie informacji niejawnych (Dziennik Ustaw z 1999 nr 11 poz. 95)

¹⁵ Ustawa z dnia 5 lipca 2002 o ochronie niektórych usług świadczonych drogą elektroniczną opartych lub polegających na dostępie warunkowym (Dziennik Ustaw z 2002 Nr 126, poz. 1068)

3. jeżeli sieć instytucji zostanie wykorzystana do czynów, za które grozi odpowiedzialność karna (choćby dystrybucja materiałów pornograficznych), możliwe będzie wskazanie osób odpowiedzialnych

1.5. Kryteria wyboru metody uwierzytelniania

W poprzednich podrozdziałach pokazano, iż rozważania o uwierzytelnianiu w instytucji edukacyjnej, a w szczególności w wydziale uczelni wyższej, nie są jedynie rozważaniami typowo akademickimi (teoretycznymi). Istnieje bowiem realna potrzeba, żeby je tam wprowadzać.

Jakie kryteria wobec tego powinna spełniać metoda uwierzytelniania, aby mogła być wykorzystana w takich miejscach ?

Można tutaj wyróżnić przynajmniej trzy:

- wiarygodność i skuteczność danego rozwiązania - jeżeli bowiem zbyt duża liczba uprawnionych osób nie będzie mogła przejść całego procesu uwierzytelniania (często jest to określane jako FRR, czyli False Rejection Rate – współczynnik fałszywego odrzucenia) albo zbyt dużo nieuprawnionych osób zostanie dopuszczonych do działania w chronionym systemie (dla wielu metod podawany jako FAR, czyli False Acceptance Rate - współczynnik fałszywej akceptacji), taka metoda będzie nieprzydatna.
- niskie koszty wdrożenia i utrzymania (konserwacji, ewentualnych materiałów eksploatacyjnych, itp.). Jest to kryterium, na które zwraca się uwagę praktycznie wszędzie i zawsze. Niewątpliwie zaletą byłaby możliwość integracji i wykorzystania rozwiązań (zarówno sprzętowych jak i programowych) już obecnych w instytucji.
- metoda powinna być możliwie przyjazna dla użytkowników (łatwa w użyciu, nie wymagająca od nich ponoszenia zbyt dużych kosztów, itp.). Inaczej po prostu nie zgodzą się na jej używanie albo będą w istotny sposób łamać pewne warunki niezbędne, aby metoda mogła poprawnie spełniać swoją rolę.

2. Charakterystyka metod i technik uwierzytelniania wykorzystywanych we współczesnych systemach informatycznych

W celu dokonania wyboru metody uwierzytelniania zgodnie z wytycznymi z podrozdziału 1.5 konieczne jest wykonanie ich pełnego przeglądu. Poniższe opracowanie składać się będzie z trzech części:

1. przedstawienia pojęć i funkcji matematycznych wykorzystywanych przy uwierzytelnianiu (podrozdział 2.1)
2. opisu wykorzystywanych schematów działania wykorzystujących m.in. wspomniane funkcje (podrozdział 2.2)
3. wyliczenia konkretnych metod, gdzie zawarte będą m.in. informacje, w połączeniu z jakim schematem można wykorzystać każdą z metod (podrozdziały 2.3 – 2.6)

W każdej z tych części opisane zostaną wynikające z przyjętych rozwiązań problemy i wskazane sposoby ich unikania lub zmniejszania. Metody uwierzytelniania ułożono względem kosztów (implementacji i używania) oraz popularności. Nie zastosowano znanego z niektórych opracowań podziału na:

1. metody wymagające zapamiętywania danych
2. sposoby związane z posiadaniem jakichś przedmiotów
3. grupę wymagającą rejestracji informacji o cechach fizycznych użytkowników

Każda z metod zostanie opisana osobno - przy takim założeniu zostaną również przedstawione wszystkie ich słabości. Wszystkie odstępstwa od tej zasady zostaną zasygnalizowane – przy właściwie każdym podziale niektóre metody zaczynają się bowiem nierozdzielnie ze sobą łączyć. Przykładowo: chociaż oczywiście istnieją wyjątki, dostęp do telefonu komórkowego GSM zazwyczaj wymaga posiadania przedmiotu, jakim jest karta SIM wraz z informacją, jaką jest kod PIN (Personal Identification Number). W przyjętym tutaj podziale z kolei można to zaklasyfikować jako metodę związaną z użyciem klawiatury, ale należy to również opisać przy okazji metod wykorzystujących specjalne przedmioty (karty chipowe).

Niezależnie od wniosków dotyczących „jakości” poszczególnych rozwiązań trzeba pamiętać o jednym fakcie – każda z tych metod (nawet „najgorsza”) będzie spełniać swoją rolę znacznie lepiej, gdy system informatyczny jest „ukryty” przed możliwie dużą liczbą osób postronnych i gdy jest on znacznie prostszy.

Jeżeli więc np. planowane jest uruchomienie dostępu bezprzewodowego do jakichś zasobów i jest to możliwe, należy całość skonfigurować tak, aby system je obsługujący „nie rozgłaszał” publicznie informacji o tym, ale raczej wyłącznie odpowiadał, gdy ktoś wyśle komplet poprawnych danych niezbędnych do uwierzytelnienia.

Z kolei rozwiązanie prostsze ma mniej potencjalnych błędów i luk. Z tego powodu należy wyłączać wszystkie niepotrzebne usługi np. w systemie operacyjnym, w którym działa podsystem uwierzytelniający użytkowników.

Warto sformułować również trzecią uniwersalną zasadę: każdy system powinien pokazywać swoim użytkownikom, kiedy byli poprzednio uwierzytelnieni z sukcesem (i jeżeli to możliwe, kiedy ostatnia próba ich uwierzytelnienia skończyła

się błędem). Dzięki temu jest duża szansa, że przynajmniej część użytkowników systemu zdoła zauważyć, iż ktoś inny skorzystał z ich danych identyfikacyjnych.

2.1. Funkcje matematyczne wykorzystywane przy uwierzytelnianiu

Systemy informatyczne operują na danych, z kolei uwierzytelnianie jest operacją, w której przekazywane są one systemom (które tylko na tej podstawie mogą zweryfikować tożsamość użytkownika). Ludziom korzystającym z systemu będzie przy tym zależało, aby żadna osoba trzecia nie mogła zobaczyć tego, co przekazują albo żeby żadna osoba trzecia nie mogła wykorzystać tego, co zobaczyła, w celu „podszycia się” pod nich.

Najbardziej naturalnym rozwiązaniem jest wykorzystanie w tym celu szyfrowania. Aby móc to zrobić i skorzystać z mających podstawy teoretyczne (matematycznych) funkcji, należy tylko zapisać przesyłane dane jako liczby (co nie jest trudne – wystarczy pobrać np. reprezentację dziesiętną kodu każdego znaku). Okazuje się, że oprócz teorii związanej z szyfrowaniem (i łamaniem szyfrów) przy takim podejściu zastosowanie znajdują również inne zdobycze matematyki.

Regułą jest, że schemat działania wszystkich uznanych za standardy funkcji tego typu jest publicznie dostępny. Pozwala to na badanie ich jakości. Przyjmuje się zresztą, że jeżeli jakaś funkcja jest utajniona, można się spodziewać w niej słabych elementów i nie powinno się jej stosować.

„Bezpieczeństwo” praktycznie każdej z nich opiera się na tym, że próby odkodowania danych nią zaszyfrowanych mogą zająć po prostu zbyt dużo czasu. Z drugiej strony zdolności obliczeniowe systemów komputerowych stale rosną i prędzej czy później czas ten zmniejsza się do akceptowalnego poziomu (np. godzin czy dni). Konieczne jest wtedy opracowanie doskonalszej metody. Wydaje się, że ten „wyścig” nie ma końca i tak naprawdę nie ma na to sposobu. Po prostu niezbędne jest śledzenie postępów w dziedzinie zabezpieczeń i stałe uaktualnianie odpowiednich fragmentów systemów.

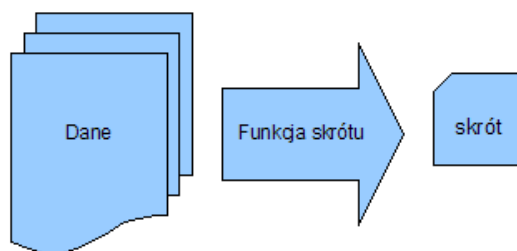
Poniżej przedstawione zostaną niektóre z tych funkcji. Nie będzie to jednak ich formalny opis, a raczej wyszczególnienie ich cech praktycznych. Gdzie to tylko będzie możliwe, podane zostaną odnośniki do dokumentów będących ich dokładną specyfikacją. Warto pamiętać, iż niektóre z opisanych rozwiązań są chronione prawnie i np. nie można ich używać do celów komercyjnych bez uiszczenia odpowiedniej opłaty.

2.1.1. Funkcje „skrótów”

Mianem funkcji „skrótów” (ewentualnie funkcji „haszujących”) określa się funkcje, które w wyniku operacji matematycznych na ciągu znaków (jego reprezentacji liczbowej) tworzą inny ciąg znaków (zwany dalej skrótem, odciskiem lub haszem) i spełniają trzy właściwości:

1. działają tylko w jedną stronę (z ciągów znaków można uzyskać hasz, ale z haszu nie można odtworzyć ciągu znaków)
2. niemożliwe jest stworzenie tego samego haszu w wyniku operacji na różnych ciągach znaków (brak jest tzw. „kolizji”)

3. z ciągów znaków o różnej długości powstaje hasz zawsze o tej samej długości (jest to ważna zaleta - z haszu nie można nic wywnioskować o długości pierwotnego tekstu).



Rysunek 4. Schemat działania jednokierunkowych funkcji skrótu (źródło: opracowanie własne)

W praktyce własność 2 i 3 się wykluczają – teoretycznie zawsze mogą istnieć takie ciągi znaków, które dadzą ten sam hasz (wynika to z podstaw matematycznych - z własności 3 widać, że hasz ma zawsze tę samą długość, a liczba kombinacji w ciągu o tej samej długości jest skończona). Z wyżej wymienionych powodów przyjmuje się więc jedynie, że dla funkcji skrótu wystarczy odpowiednio niskie prawdopodobieństwo wystąpienia sytuacji z punktu 2, aby punkt ten był spełniony. Podobnie uznaje się, że własność 1 jest również prawdziwa dla funkcji, jeżeli wygenerowanie ciągu znaków z haszu jest zbyt złożone obliczeniowo (wymaga zbyt dużego czasu).

Najbardziej chyba znaną rodziną funkcji tego typu są funkcje z rodziny MD (Message Digest) - MD2 (RFC 1319), MD4 (RFC 1320) i MD5 (RFC 1321). Były one kolejno tworzone przez profesora MIT Ronalda Linn Rivesta na przełomie lat 80 i 90 XX wieku. W chwili obecnej MD2 i MD4 nie spełniają już własności 2, podobnie pojawiają się informacje o możliwości uzyskania kolizji w MD5¹⁶. Warto dodać, że ostatnia funkcja jest popularna z innego powodu – w wystarczającym stopniu nadaje się do sprawdzania poprawności pobierania plików z Internetu (jeżeli skrót wygenerowany MD5 z pobranego zbioru jest taki jak podaje podmiot publikujący plik, przyjmuje się, że wszystko jest z nim w porządku).

Innymi funkcjami haszującymi są funkcje z rodziny SHA (Secure Hash Algorithm) – SHA-1, SHA-224, SHA-256, SHA-384 i SHA-512. Wszystkie zostały stworzone przez amerykańską NSA (National Security Agency). Informację o pierwszej opublikowano w 1993, opis kolejnych pojawił się na początku XXI wieku (2001-2002). Funkcje SHA-224 – SHA-512 są czasem określane mianem SHA-2. Przykłady implementacji wszystkich funkcji podano w RFC 4634 oraz (z wyłączeniem SHA-224) w FIPS 180, SHA-1 jest opisano również we wcześniejszym RFC 3174.

Nie można tutaj natomiast dodać funkcji z rodziny CRC (Cyclic Redundancy Check) z uwagi na zbyt łatwą możliwość znalezienia „kolizji”. Podobnie jest dla różnych metod generowania sum kontrolnych.

Istnieją już firmy, które tworzą zestawy skrótów dla różnych popularnych funkcji „haszujących” dla kolejnych możliwych ciągów znaków w oparciu o metodę

¹⁶ Xiaoyun Wang i Hongbo Yu, „How to Break MD5 and Other Hash Functions”, Shandong University, Jinan 250100, luty 2005, Chiny (<http://www.infosec.sdu.edu.cn/paper/md5-attack.pdf>)

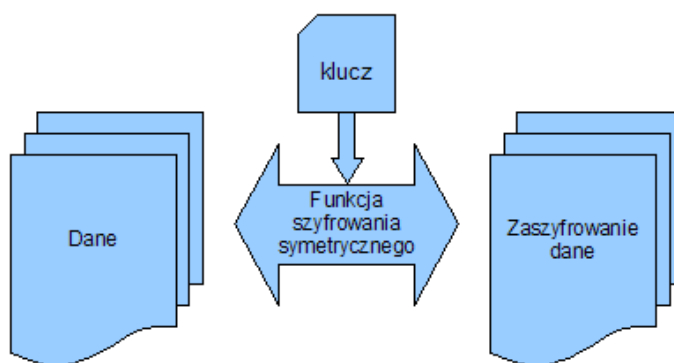
Philippe Oechslin¹⁷ i sprzedają je (lub możliwość ich użycia) w Internecie. Zestawy te są nazywane tzw. „tęczowymi tablicami” (rainbow tables). Chociaż zawierają jedynie ułamek wszystkich możliwych kombinacji skrót-ciąg, to pozwalają na szybkie „odszyfrowanie” ciągów zawierających określone znaki („Zwykle bazy danych hashów zajmują setki gigabajtów, przez co są nieefektywne. Tęczowa tablica jest tworzona przez zapisywanie łańcuchów (ang. chains) ze skrótów z możliwych haseł. Dzięki temu zapisywany jest jeden hash na kilkaset, a nawet kilka tysięcy wygenerowanych, a mimo to baza danych pozwala na odwrócenie hasha w ponad 90% przypadków, w zależności od rozmiaru tablicy. Czas łamania hasła skraca się przez to do kilkudziesięciu sekund na silnych komputerach.”¹⁸). Obroną przed użyciem takich tablic jest:

- wykorzystywanie możliwie niestandardowych znaków w treści ciągów
- wykorzystywanie takich funkcji haszujących, które oprócz ciągów do kodowania wymagają podania także jakiejś liczby losowej (tzw. „soli” – po angielsku „salt”). Liczba jest wykorzystywana we wzorze funkcji haszującej, a następnie zapisywana obok haszu – ponieważ tablice są tworzone dla konkretnej funkcji haszującej, konieczne byłoby przygotowanie wartości dla każdej możliwej wartości takiej liczby losowej, co mogłoby być niewykonalne obliczeniowo.

Przykładem programu wykorzystującego „tęczowe tablice” jest Ophcrack¹⁹, przykładem serwisu sprzedającego usługi z ich wykorzystaniem jest Decryptum²⁰.

2.1.2. Szyfrowanie symetryczne

Szyfrowanie symetryczne wykorzystuje jeden ciąg znaków (klucz) do szyfrowania danych i ten sam ciąg (lub ciąg znaków możliwy w łatwy sposób do uzyskania z klucza szyfrującego) do ich odszyfrowania. Możemy wyróżnić szyfry strumieniowe (które operują na kolejnych bajtach ciągu danych) i blokowe (które dzielą dane na bloki o określonej długości i operują na tych blokach).



Rysunek 5. Schemat działania szyfrów symetrycznych
(źródło: opracowanie własne)

¹⁷ Philippe Oechslin, „Making a Faster Cryptanalytic Time-Memory Trade-Off”, Laboratoire de Sécurité de Cryptographie (LASEC) Ecole Polytechnique Fédérale de Lausanne, Szwajcaria, maj 2003 (<http://lasecwww.epfl.ch/~oeechslin/publications/crypto03.pdf>)

¹⁸ Wikipedia (http://pl.wikipedia.org/wiki/T%C4%99czowe_tablice)

¹⁹ <http://ophcrack.sourceforge.net/>

²⁰ <http://www.decryptum.com/>

Należy zwrócić tutaj uwagę na jedną rzecz: dane zakodowane w ten sposób nie będą mogły być odczytane przez osoby trzecie, jeżeli te nie poznają klucza (z tego powodu należy go bezwzględnie chronić). Gdy zaś klucze będą tworzone w sposób przewidywalny, nawet zastosowanie najlepszego szyfru symetrycznego nic da. Z tego powodu w systemie informatycznym konieczne (krytyczne) jest stosowanie odpowiednio skomplikowanych programowych generatorów liczb losowych, dlatego też spotyka się również rozwiązania z generatorami sprzętowymi.

Klasycznym i najprostszym przykładem szyfru strumieniowego jest funkcja XOR (zastosowana raz na danych i kluczu powoduje ich zaszyfrowanie, zastosowana ponownie na danych zaszyfrowanych i kluczu „przywraca” pierwotną treść). Oprócz prostoty jej poważną wadą jest fakt, że wykonana na danych zakodowanych i danych do kodowania daje w wyniku klucz.

Tabela 1. Schemat działania funkcji XOR na bitach danych i klucza (źródło: opracowanie własne)

Dane do kodowania	Klucz	Dane zakodowane	Klucz	Dane odkodowane
0	0	0	0	0
0	1	1	1	0
1	0	1	0	1
1	1	0	1	1

Szyfry tego typu wykorzystywane są np. w telefonii GSM do kodowania transmisji między terminalem (telefonem) i stacją bazową (BTS – Base Transceiver Station). Mowa tutaj o funkcjach A5/1 i A5/2.

Pewnym problemem w przypadku (przynajmniej niektórych) szyfrów strumieniowych jest kwestia wykrywania i zabezpieczenia kodowanych danych przez przestawianiem/zamianą bitów czy też bajtów. Niezbędne jest stosowanie dodatkowo sprawdzania integralności danych.

Przykładowo: gdyby zaszyfrowano wyłącznie funkcją XOR np. jakieś informacje identyfikacyjne, osoba atakująca tak naprawdę nie byłaby zmuszona do odkodowania ciągu, aby móc wstawić tam dane związane z sobą. Stąd system korzystający z takiego szyfrowania musi jeszcze używać dodatkowych mechanizmów kontrolnych.

Inaczej działają symetryczne szyfry blokowe. Jednym z wczesnych (I połowa lat 70 XX wieku) rozwiązań jest DES (Data Encryption Standard). Przez kilkanaście lat (od 1977 do 2005) był on standardem używanym do kodowania informacji rządowych w USA. Opisywano go w dokumentach oznaczonych jako FIPS PUB 46 – ostatnia wersja FIPS PUB 46-3 pochodzi z 1999.

Operuje on na 64-bitowych blokach danych. Szyfrowanie rozpoczyna się od wykonania operacji matematycznej zwanej permutacją wstępną i podziału bloku danych na dwie części (każda wielkości 32 bitów).

Następnie powtarzane są 16 razy (w tzw. rundach) następujące czynności: na jednej z części (można ją nazwać drugą) wykonuje się tzw. funkcję Feistela, a następnie na otrzymanym wyniku i pierwszej części funkcję XOR. To, co zostanie wyliczone, staje się drugą częścią wejściową dla kolejnej rundy (jako pierwszą bierzemy tam drugi blok z rundy obecnej).

Ostatnią operacją jest połączenie otrzymanych po ostatniej rundzie bloków i wykonanie operacji matematycznej zwanej permutacją końcową.

Klucz w tym algorytmie składa się z 64 bitów (przy czym jedynie 56 jest wykorzystywane do kodowania). Jego części są wykorzystywane w funkcji Feistela w kolejnych rundach.

Pod koniec lat 90 XX (1998 – 1999) wieku istniał już możliwy do przygotowania stosunkowo niskim kosztem odpowiedni sprzęt i oprogramowanie (przykładowo – maszyna DES Cracker skonstruowana przez Electronic Frontier Foundation), które metodą „brutalnej siły” (tzw. atak siłowy, czyli „brute force attack”, który w tym przypadku polega na szyfrowaniu wszystkich możliwych wiadomości i porównywaniu ich z zakodowanym wzorcem) potrafiło rozkodować zaszyfrowane DESem wiadomości w przeciągu mniej niż 24h.

Istniały próby usunięcia ograniczeń DES np. poprzez trzykrotne szyfrowanie nim danych (rozwiązanie było nazywane 3DES, Triple DES, TDES albo TDEA – Triple Data Encryption Algorithm), jednakże nie przyjęły się szerzej (były to półśrodki nie eliminujące rzeczywistych przyczyn słabości DES).

Następcą funkcji DES został AES (Advanced Encryption Standard) znany też jako Rijndael, którego opis można znaleźć w FIPS PUB 197 (rok 2001).

Kolejnym popularnym schematem szyfrowania symetrycznego jest Blowfish (stworzony przez Bruce Schneiera w 1993)²¹.

W wielu pozycjach literatury związanej z szyfrowaniem wspomina się również o funkcji IDEA (International Data Encryption Algorithm). Została stworzona na początku lat 90 XX wieku, jednakże nie przyjęła się szerzej ze względu na ograniczenia patentowe.

Warto przypomnieć, że istota funkcji blokowych polega na operowaniu na porcjach danych. Można istotnie zmniejszyć korzyści wynikające z szyfrowania skracając długość bloków. Podobnie nie powinno się zmniejszać zakresu możliwych do zakodowania znaków. Zostało to zrobione np. w protokołach uwierzytelniania LM i NTLM wykorzystywanych we wcześniejszych wersjach Windows.

W LM znaki w ciągach są zamieniane na duże litery. Następnie (już w obu protokołach) ciągi krótsze niż 14 znaków są uzupełniane spacjami, dzielone na bloki 7 bajtowe i dopiero (oddzielnie) szyfrowane DESem. Osoba atakująca po otrzymaniu zakodowanego ciągu może podzielić go i zająć się poszczególnymi częściami oddzielnie. Ponieważ na końcu pojawiają się najczęściej spacje, częściowo (w LM) wyłącznie duże litery, wystarczy generować kolejne możliwe kombinacje tych znaków, szyfrować DESem i porównywać z poszczególnymi częściami. Istotnie zmniejsza się liczbę koniecznych do sprawdzenia kombinacji, co stawia pod znakiem zapytania bezpieczeństwo tych protokołów (dlatego zresztą Microsoft już je zastąpił nowszymi wersjami). Szerzej zostało to przedstawione np. w artykułach Marcina Szeligi²².

2.1.3. Szyfrowanie asymetryczne

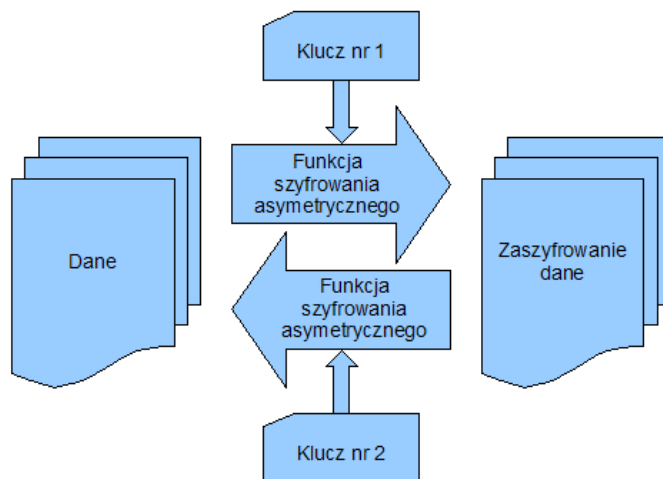
Szyfrowanie asymetryczne wykorzystuje jeden ciąg znaków (klucz) do szyfrowania danych i drugi ciąg znaków (drugi klucz) do ich odszyfrowania. W zastosowanych algorytmach (podobnie jak w szyfrowaniu symetrycznym) również

²¹ więcej pod adresem <http://www.schneier.com>

²² Marcin Szeliga, „NTLM cz.1” i „NTLM cz.2”, Microsoft, maj 2003 (<http://www.microsoft.com/poland/technet/article/art001.mspx> i <http://www.microsoft.com/poland/technet/article/art003.mspx>)

okazuje się bardzo ważne, aby wykorzystywane liczby losowe były niemożliwe do przewidzenia.

Algorytmy szyfrowania asymetrycznego można wykorzystać do dwóch celów – szyfrowania danych tak, aby mogły być odczytane tylko przez konkretną osobę (przykład jak w opisie algorytmu RSA poniżej - klucz nr 1 to klucz publiczny, klucz nr 2 to klucz prywatny) oraz sprawdzenia, czy znane dane zostały zaszyfrowane przez konkretną osobę (przykład jak w podrozdziale 2.2.3 - wtedy klucz nr 1 to klucz prywatny, klucz nr 2 to klucz publiczny).



Rysunek 6. Schemat działania funkcji szyfrowania asymetrycznego (źródło: opracowanie własne)

Jednym z najbardziej znanych algorytmów tego typu jest RSA (nazwa to pierwsze litery nazwisk twórców będących zarazem profesorami MIT – Rona Rivesta, Adi Shamira i Lena Adlemana) zwany niekiedy algorytmem MIT (np. w tłumaczeniu z „*Computer Networks*” Andrew S. Tanenbaum²³) opublikowany pod koniec lat 70 XX wieku. RSA był chroniony patentem w USA (patent nr 4 405 829) w latach 1983 – 2000.

Podany zostanie schemat działania tego algorytmu przy kodowaniu danych, które mają zostać przekazane później konkretnej osobie (po lewej stronie algorytm, po prawej konkretny przykład zaczerpnięty z angielskiej Wikipedii²⁴):

- | | |
|---|--------------|
| 1. Wybranie dwóch (możliwie dużych) liczb pierwszych p i q | $p=61, q=53$ |
| 2. Wyliczenie $n = p \cdot q$ | $n=3233$ |
| 3. Wybranie losowo liczby e , która musi być mniejsza niż n i względnie pierwsza z liczbą $(p-1) \cdot (q-1)$ (ich największym wspólnym dzielnikiem jest 1). Para liczb e i n stanowi klucz publiczny | $e=17$ |
| 4. Wyliczenie liczby d : $d = e^{-1} \bmod (p-1) \cdot (q-1)$. Para liczb d i n będzie stanowił klucz prywatny | $d=2753$ |
| 5. W celu zaszyfrowania należy podzielić wiadomość na bloki m_i o długości mniejszej niż n | $m=123$ |

²³ „*Sieci komputerowe*”, ISBN 83-204-0964-0, Wydawnictwo Naukowo-Techniczne, Warszawa 1988 (tłumaczenie dokonane przez Marian Suskiewicz, Janusz Piela, Waldemar Borkowski, Bogna Znojkiwicz-Ozyp z „*Computer Networks*” Andrew S. Tanenbaum)

²⁴ <http://en.wikipedia.org/wiki/Rsa>

każdy i na każdym bloku wykonać operację $c_i =$

$m_i^e * (\text{mod } n)$ (jak widać, wymagana jest znajomość klucza publicznego) $c = 123^{17} \text{ mod } 3233 = 855$

6. Odszyfrowanie polega na wykonaniu operacji $m_i = c_i^d * (\text{mod } n)$ (z użyciem klucza prywatnego) $m = 855^{2753} \text{ mod } 3233 = 123$

W chwili obecnej wykorzystuje się również DSA (Digital Signature Algorithm) będący standardem rządu amerykańskiego (dokument FIPS PUB 186, patent USA nr 5 231 668). Używa on funkcji haszujących z rodziny SHA.

W algorytmach tego typu całe „zabezpieczenie” danych polega na tym, iż odtworzenie klucza prywatnego z publicznego jest zbyt złożone obliczeniowo. Przykładowo: w RSA wybiera się możliwie bardzo duże liczby pierwsze p i q . Wtedy pojawia się trudność w rozłożeniu liczby n z klucza publicznego na p i q w skończonym czasie (a co za tym idzie, wyliczenia d na podstawie e).

Oczywiście możliwości systemów komputerowych cały czas rosną i stąd muszą zwiększać się również długości wykorzystywanych kluczy - m.in. z tego powodu można zaryzykować stwierdzenie, iż algorytmy asymetryczne będą zawsze wolniejsze w użyciu (będą wymagać większej ilości operacji matematycznych) niż symetryczne. Istnieje sposób na eliminację tej wady (zostanie on przedstawiony w podrozdziale 2.2.3).

2.2. Schematy przeprowadzania procesu uwierzytelniania w systemach informatycznych

Jak widać z poprzednich punktów, dostępne jest dużo środków technicznych pozwalających na ukrywanie tego, co użytkownicy chcieliby przekazać do systemu. Warto się teraz zastanowić nad tym, jak można to wykorzystać praktycznie. Zostanie to pokazane w opisie kolejnych schematów uwierzytelniania wykorzystywanych w działających systemach informatycznych.

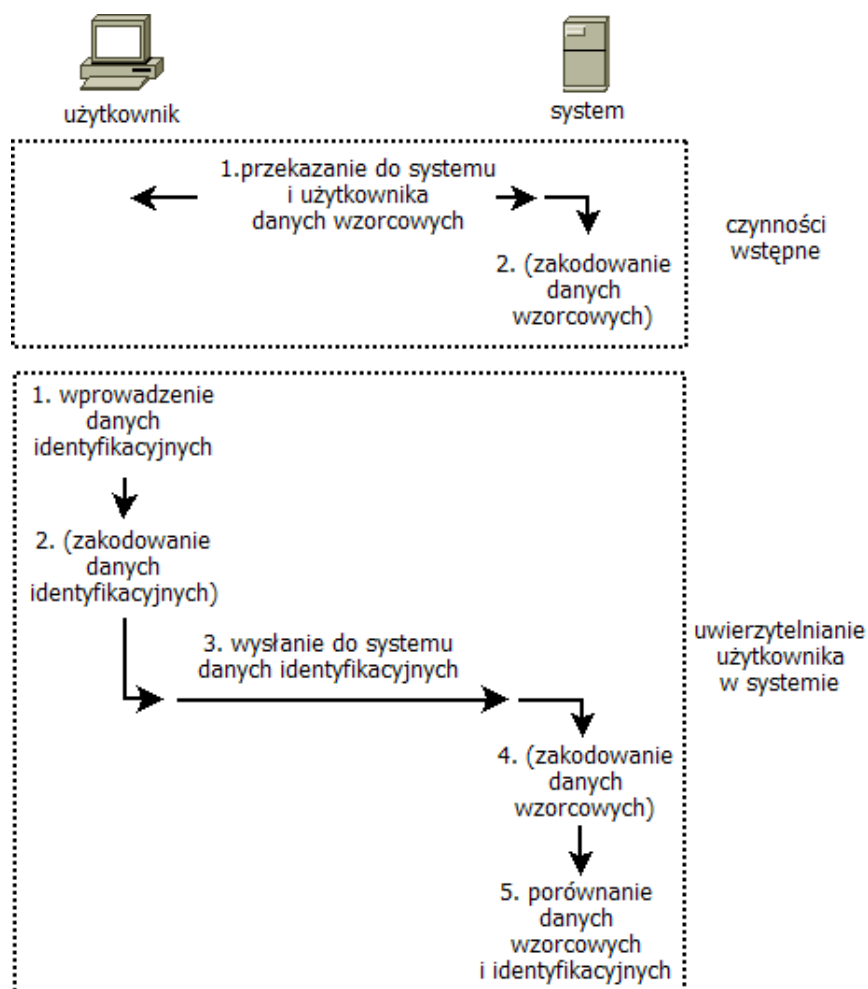
2.2.1. Porównywanie z danymi wzorcowymi

Pierwszym krokiem jest przygotowanie zestawów danych identyfikacyjnych w postaci cyfrowej. Każdy zestaw musi być unikalny i jednoznacznie przypisany do jednego użytkownika (który ma być później uwierzytelniany). Rodzaj przechowywanych danych zależy od przyjętej metody.

W trakcie działania systemu osoba uwierzytelniana musi wprowadzić drugi zestaw danych. Jest on porównywany w ściśle określony sposób ze wszystkimi przechowywanymi danymi wzorcowymi. Jeżeli z określoną dokładnością pasuje do którejś z nich, użytkownik jest uwierzytelniany jako osoba przypisana w systemie do pasujących danych wzorcowych. Pojawiają się tutaj od razu różne problemy:

1. brak jest możliwości sprawdzenia "tożsamości" systemu przez użytkownika
2. konieczność zabezpieczenia danych wzorcowych (w trakcie wprowadzania oraz przechowywania) przed niepożądanym dostępem
3. konieczność takiego zapamiętania i wprowadzania danych identyfikacyjnych przez użytkowników systemu, aby nie były dostępne dla osób nieupoważnionych

4. konieczne jest sensowne ustalenie, z jaką dokładnością należy porównywać wzorce z danymi podawanymi przez uwierzytelnianych użytkowników. Problem też można również czasem sformułować inaczej – jakie teoretyczne ryzyko związane z tym, że system poprawnie uwierzyteli osobę nieupoważnioną i niepoprawnie uwierzyteli osobę upoważnioną, można zaakceptować ?



Rysunek 7. Najprostszy schemat uwierzytelniania wykorzystuje tylko porównywanie z danymi wzorcowymi (źródło: opracowanie własne)

Część danych można poddać dodatkowo działaniu funkcji „skrót” (dalej otrzymany wynik będzie nazywany haszem). Pewne korzyści wynikające z tego mogą dać dwa rozwiązania:

1. w systemie jako hasz przechowywane są wyłącznie dane wzorcowe, dane od użytkownika są poddawane działaniu funkcji „skrót” po ich otrzymaniu
2. w systemie przechowuje się dane wzorcowe w postaci „haszu”, dane użytkownika są zapisywane w tej postaci przed ich przesłaniem do systemu

W obu uzyskaną zaletą jest to, iż w przypadku włamania do systemu uzyskane wersje zakodowane mogą być bezużyteczne dla włamywacza. W rozwiązaniu drugim dodatkowa korzyść wynika z tego, że dane nie są jawne na drodze między użytkownikiem i systemem (co może być istotne, gdy są przesyłane poprzez Internet).

Wykorzystanie funkcji skrótu stwarza pewne utrudnienia:

1. wymagana jest pewna moc obliczeniowa do wykonania operacji matematycznych związanych z funkcjami skrótu
2. jeżeli użytkownik zapomni swoich danych identyfikacyjnych, osoba upoważniona nie jest mu w stanie podać ich wersji odkodowanej (w takiej sytuacji można jedynie wygenerować nowy zestaw danych i zapisać go w postaci zakodowanej)

Stosuje się go jednak dosyć powszechnie. Utrudnienie drugie jest bowiem również zaletą – jeżeli ktoś (np. włamywacz) ma jedynie dostęp do wersji zakodowanych (przykładowo: uzyskał nieautoryzowany dostęp do odpowiedniego pliku na serwerze www) i nie możliwości modyfikacji systemu, musi próbować generować kolejne możliwe dane identyfikacyjne i kodować je funkcją haszującą, aby móc porównać je z danymi już zakodowanymi i uzyskać w ten sposób informację, co należy podać systemowi. Podchodząc do tego praktycznie: jeżeli w tej formie przechowywane są np. hasła użytkowników systemu, wystarczy zastosować zasady opisane w podrozdziale „hasła wielokrotne”, aby dosyć skutecznie zwiększyć liczbę wymaganych prób (co z kolei może zniechęcić włamywacza). Rozważania tego typu nie są niestety tylko teoretyczne – informacje o tym, jak z użyciem Google uzyskać dostęp do plików tego typu (które teoretycznie powinny być niedostępne), pojawiały się już nawet w popularnej prasie komputerowej²⁵.

2.2.2. „Wyzwanie - odpowiedź”

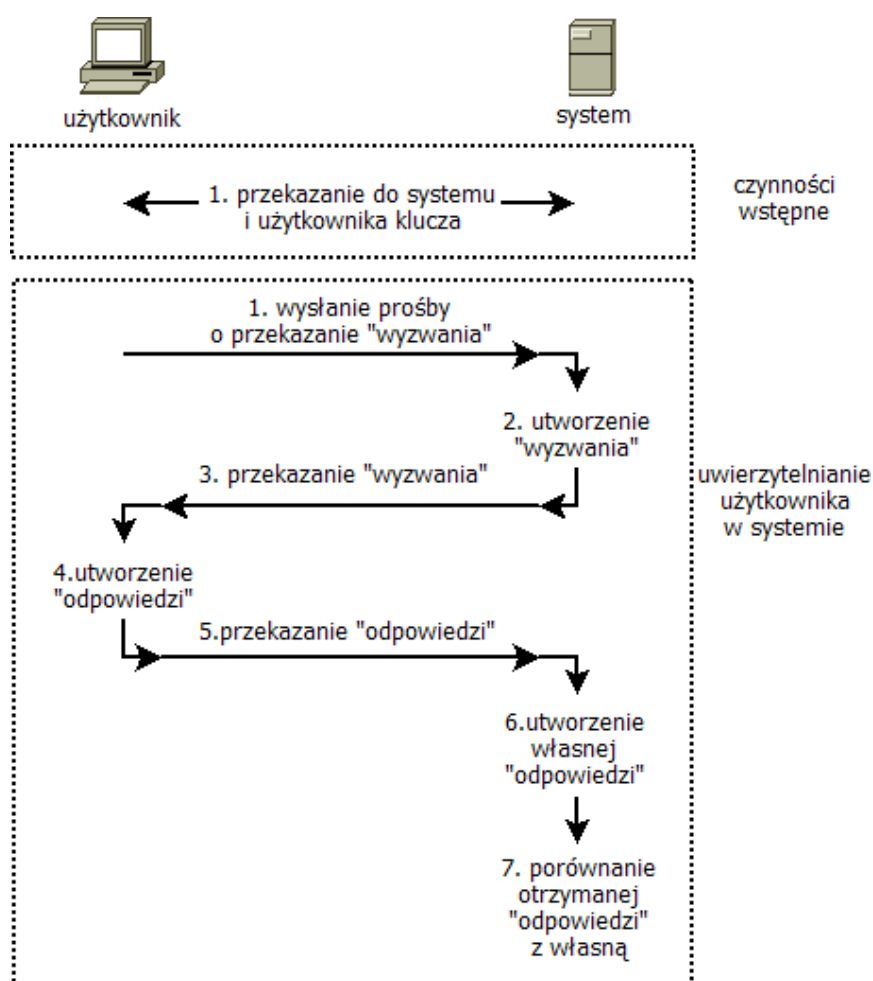
W tym schemacie (zwanym również w różnych publikacjach „uwierzytelnianiem współdzielonych kluczy”) użytkownik jak i system muszą posiadać ten sam ciąg znaków (zwany dalej właśnie kluczem). Istota tego rozwiązania polega na tym, iż sam klucz (ani jego skrót) nie jest w ogóle przekazywany do systemu w trakcie uwierzytelniania. Najbardziej ogólny schemat postępowania wygląda następująco:

1. użytkownik wysyła do systemu prośbę o wysłanie danych. Będą one tzw. „wyzwaniem” („challenge”).
2. system wysyła do użytkownika „wyzwanie”.
3. użytkownik szyfruje otrzymane dane z użyciem swojego klucza i odsyła do systemu (jest to „odpowiedź” – „response”).
4. system porównuje to co otrzymał z tym, co otrzyma po zakodowaniu „wyzwania” swoim kluczem (jeśli dane te będą identyczne, będzie miał potwierdzenie, iż użytkownik korzysta z tego samego klucza).

Całość można powtórzyć w odwrotną stronę (to użytkownik wysyła „wyzwanie”) i wtedy również użytkownik będzie miał pewność, iż system używa tego samego klucza. W tym schemacie ważne jest, aby wyzwania były losowe (inaczej osoba atakująca mogłaby być uwierzytelniona po przekazaniu raz przechwyconej „odpowiedzi”). Powoduje to, że konieczne jest przechowywanie

²⁵ Filip Zagórski, „Wyszukiwanie tajnych informacji w Google'u”, Chip 2/2006, str. 113 (http://www.chip.pl/arts/archiwum/n/articlear_165153.html) lub Paweł Brągoszewski, „Zostań hackerem w weekend”, PC World Komputer 4/2007, str. 86 (<http://www.pcworld.pl/artykuly/54527.html>)

wysłanych „wyzwań” i stosowanie środków pozwalających na stwierdzenie, że dany użytkownik otrzymał konkretne „wyzwanie”.



Rysunek 8. Schemat uwierzytelniania „wyzwanie-odpowiedź”
(źródło: opracowanie własne)

Tego typu schemat został zastosowany np. w systemach Windows (protokół NTLM²⁶), jest wykorzystywany w standardzie Bluetooth, jak również w sieciach WiFi (szyfrowanie WEP - Wired Equivalent Privacy).

Warto zauważyć, iż problemem jest sytuacja, gdy osoba trzecia uzyska (np. „podsluchując”) odpowiednią liczbę „wyzwań” i „odpowiedzi” i na ich podstawie odtworzy klucz. Należy tak konstruować algorytmy, aby było to jak najbardziej złożone obliczeniowo, ale tak naprawdę jedyną obroną przed tym jest jak najczęstsze zmienianie kluczy. Poniżej zostanie pokazane, jak tego typu spostrzeżenia wykorzystano do przełamania właśnie wspomnianego szyfrowania WEP.

W tym systemie klucz jest tworzony na podstawie haseł wpisywanych do urządzeń przez użytkowników. W trakcie transmisji generowana jest liczba losowa (o maksymalnej wielkości 2^{24} , co okazało się w praktyce wartością zbyt małą), z klucza i liczby tworzony jest ciąg (zwany dalej kodowym), który jest wraz z przesyłanymi przez użytkownika danymi poddawany działaniu funkcji XOR. Żeby

²⁶ przedstawiony np. w Bazie Wiedzy (Knowledge Base) Microsoftu pod numerem 102 716 i artykułach Marcina Szeligi

można było po drugiej stronie połączenia odczytać zaszyfrowany ciąg, konieczne jest przesłanie tam z nim otwartym tekstem również wybranej liczby losowej.

Osoba atakująca po odebraniu drogą radiową zaszyfrowanego ciągu musi go poddać działaniu funkcji XOR ze znanym tekstem (może to być np. przesłany przez nią do sieci email albo fragmenty tekstów używane w typowych protokołach sieciowych) otrzymując ciąg kodowy. Ponieważ liczba losowa jest znana (była również odebrana drogą radiową), można odczytać klucz.

Oczywiście schemat ten wymaga analizy stosunkowo dużych ilości danych (różne źródła podają informacje o wielkości rzędu 160 MB), niemniej jednak jest możliwy do przeprowadzenia i istnieją gotowe narzędzia pozwalające na jego praktycznie automatyczne wykorzystanie (takie jak Aircrack).

W praktyce okazuje się niestety, że problem braku odpowiedniego zabezpieczenia sieci bezprzewodowych jest jeszcze poważniejszy. Około 42% użytkowników nie stosuje w ogóle bowiem **ŻADNEGO** szyfrowania (tak można interpretować wynik eksperymentu pokazany na rysunku 9, a konkretnie brak informacji w materiale źródłowym o tym, że do kategorii „sieci bez WEP” zaliczono sieci z szyfrowaniem innym niż WEP).



Rysunek 9. Wynik eksperymentu Kaspersky Lab sprawdzającego zabezpieczenia sieci WiFi²⁷ (dane z Warszawy z kwietnia 2007, podobnie jest w innych krajach)

W przypadku Bluetooth można z kolei zauważyć, że praktycznie wszystkie rodzaje ataków (przedstawione na stronie Trinite²⁸) nie są związane ze słabościami użytego schematu uwierzytelniania, ale raczej z błędami dotyczącymi użytych protokołów transmisji (jakich jak OBEX) albo wykorzystują (metoda Car Whisperer) fakt używania przez producentów urządzeń zawsze tego samego klucza (w rodzaju 0000 lub 1234), który nie jest (albo nie może być) po prostu zmieniony przez użytkowników.

2.2.3. Uwierzytelnianie za pomocą kluczy publicznych

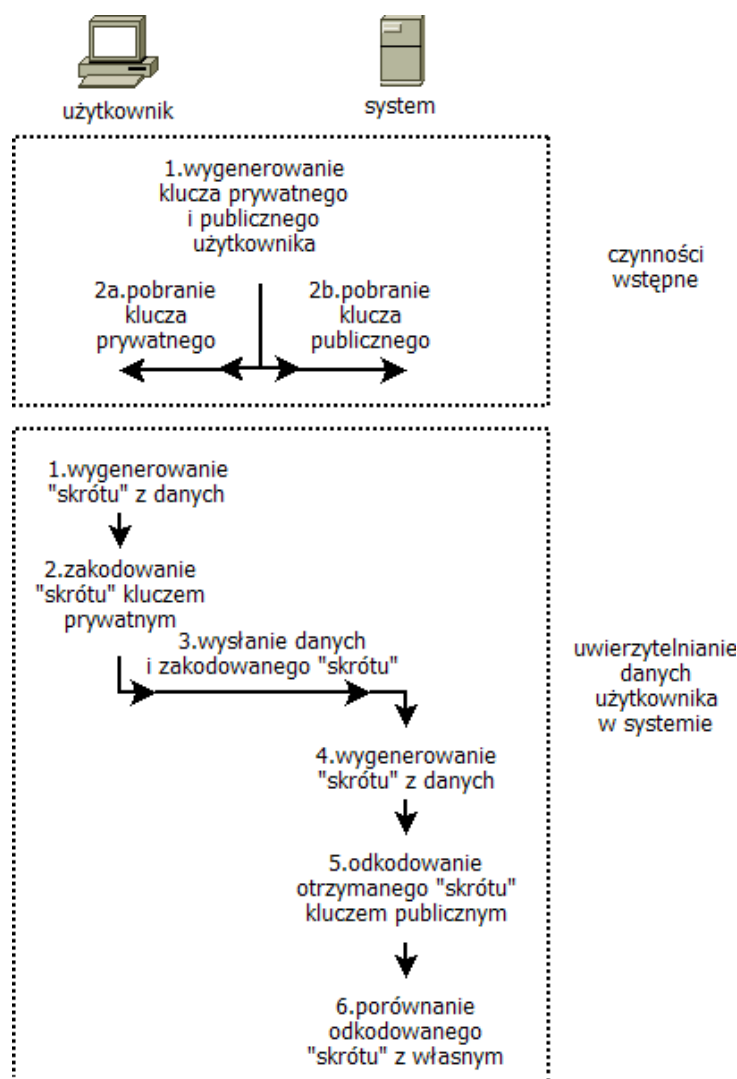
Dwa schematy, które zostaną teraz podane, zostały już właściwie częściowo przedstawione i zasygnalizowane przy okazji opisu szyfrowania asymetrycznego.

²⁷ <http://viruslist.pl/analysis.html?newsid=423>

²⁸ <http://trifinite.org>

W pierwszym z nich użytkownik musi posiadać swój klucz prywatny i publiczny. Jeżeli chce wysłać do systemu dane w formie jawnej (zwane dalej dokumentem) tak, aby można było sprawdzić ich autentyczność, musi wykonać następujące kroki:

1. wywołać funkcję „skrót” na dokumencie
2. zaszyfrować uzyskany „skrót” swoim kluczem prywatnym (otrzymany ciąg nazywany jest często podpisem elektronicznym)
3. wysłać dokument wraz z uzyskanym w poprzednim punkcie ciągiem



Rysunek 10. Schemat uwierzytelniania wykorzystywany m.in. w podpisie elektronicznym (źródło: opracowanie własne)

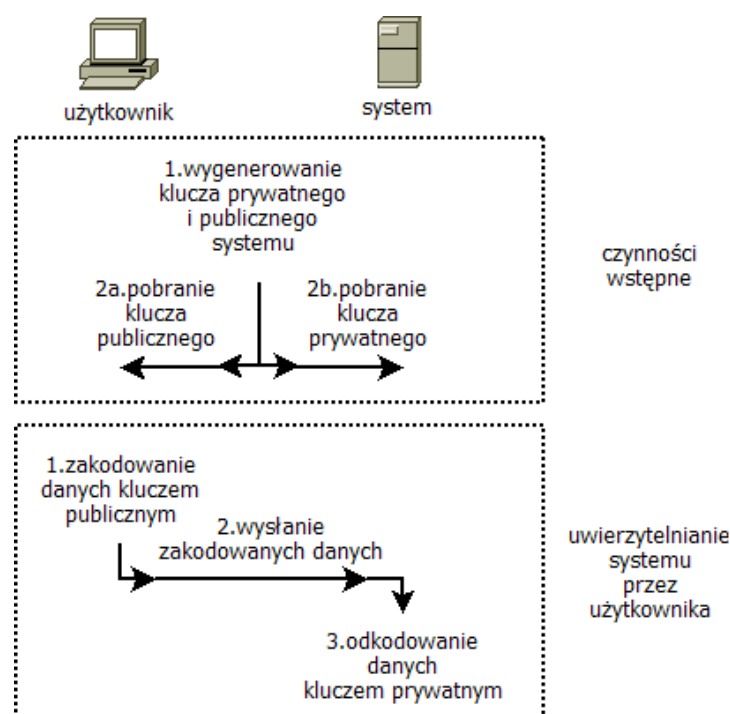
Aby system mógł sprawdzić tożsamość osoby wysyłającej dokument, po jego otrzymaniu (z załączonym ciągiem) musi:

1. wywołać funkcję „skrót” na dokumencie
2. deszyfrować dołączony do dokumentu ciąg kluczem publicznym osoby wysyłającej

3. porównać uzyskany w punkcie 2 ciąg z wartością uzyskaną z funkcji skrótu – jeżeli są identyczne, można przyjąć tożsamość użytkownika, do którego należał klucz publiczny, za potwierdzoną.

Drugi schemat służy do przeprowadzenia pełnego kodowania danych przy przesyłaniu ich do systemu. Podobnie jak poprzednio najpierw wymagane jest wygenerowanie klucza prywatnego i publicznego (ale tym razem systemu). Użytkownik chcąc komunikować się z nim w sposób bezpieczny musi:

1. wygenerować klucz do szyfrowania symetrycznego
2. zakodować klucz z punktu 1 kluczem publicznym systemu
3. wysłać zakodowany klucz z punktu 2 do systemu



Rysunek 11. Najbardziej ogólny schemat uwierzytelniania używanego np. w SSL (źródło: opracowanie własne)

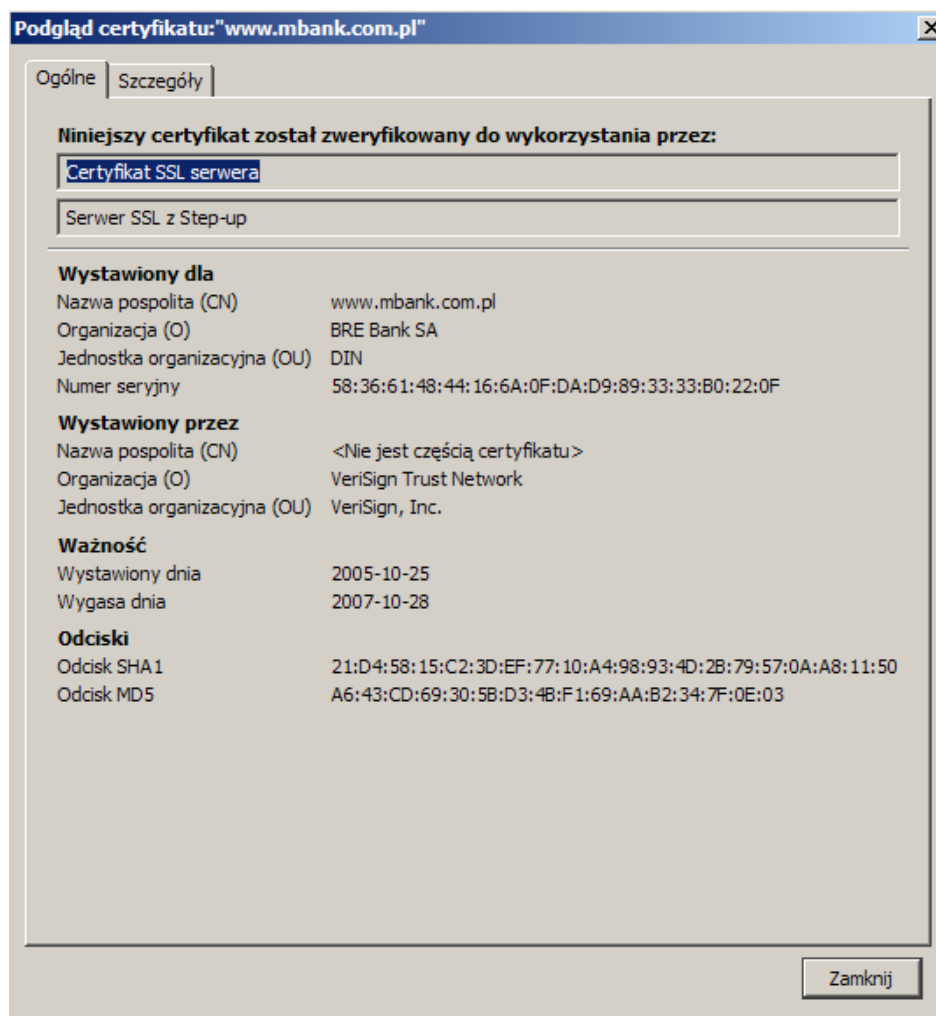
System może odszyfrować otrzymany klucz i od tej chwili transmisja może być kodowana stosunkowo szybkim algorytmem symetrycznym. Ważną zaletą całości jest fakt, że żadne dane identyfikacyjne nie są w ogóle przesyłane w formie jawnej. Rozwiązanie tego typu stosowane jest (oczywiście z elementami dodatkowymi) m.in. w popularnym protokole SSL (Secure Socket Layer) wykorzystywanym np. w przeglądarkach www. Można oczywiście stwierdzić, że zastosowano „niepotrzebne” komplikacje i wystarczy tutaj zastosować sam algorytm asymetryczny tak jak zostało to przedstawione w podrozdziale 2.1.3 przy opisie RSA. Jest to prawda – chodzi tutaj jednak również o to, żeby całość była jak najszybsza, a algorytmy symetryczne są z reguły szybsze niż asymetryczne (dlatego też oprócz niezbędnej pierwszej fazy związanej z uwierzytelnieniem dalej stosowany będzie tylko ten pierwszego rodzaju).

W rozwiązaniu tym oczywiście tylko użytkownik uzyskuje pewność, że wysyła dane do prawdziwego systemu (założenie to opiera się na teorii szyfrów asymetrycznych). Żeby również serwer „wiedział”, od kogo odbiera dane, można

np. do informacji przesyłanych w punkcie 2 dodać dokument podpisany cyfrowo przez użytkownika (jak to opisano na początku tego podrozdziału).

Oba schematy (związany z podpisem i SSL) mają sens jedynie wtedy, gdy można w wiarygodny sposób stwierdzić, że właścicielem określonego klucza jest konkretna osoba lub instytucja. Czasami można to rozwiązać w sposób nie budzący wątpliwości (np. w małej firmie klucze będą rozdawane przez jednego człowieka, który będzie prowadził odpowiednią ewidencję i udostępniał takie dane). Nie zawsze jest to jednak tak proste i możliwe. Z tych powodów powstały różne firmy takie jak VeriSign czy Certum (w Polsce mowa jest o urzędach certyfikacji). Są one częścią tzw. infrastruktury klucza publicznego (PKI - Public Key Infrastructure). Zajmują się bowiem (na skalę globalną) tworzeniem i wydawaniem kluczy prywatnych (w sposób, który ma z założenia uniemożliwić ich przekazanie osobom trzecim) oraz udostępnianiem publicznych. W Polsce instytucje tego typu działają m.in. w oparciu o w ustawę o podpisie elektronicznym²⁹.

Każdy klucz jest przez nie powiązany m.in. z informacjami o właścicielu oraz datą ważności (mowa jest wtedy o tzw. certyfikacie). Jeżeli certyfikat jest zgodny z normą ISO/IEC 9594-8 (zwaną również X.509), zawiera również dane o organie wydającym (zakodowane kluczem prywatnym tego organu) oraz sposobie wydania.



Rysunek 12. Przykład danych certyfikatu w przeglądarce Mozilla Firefox

²⁹ Ustawa z dnia 18 września 2001 o podpisie elektronicznym (Dziennik Ustaw z 2001 Nr 130, poz. 1450)

Często mówi się o tzw. ścieżce certyfikacji – jest to hierarchiczna informacja o podmiotach, których certyfikaty zostały użyte do poświadczenia autentyczności danego certyfikatu. Przykładowo: jeżeli certyfikat Jana Kowalskiego został wydany przez firmę X (której certyfikat wydała z kolei firma Y, a firmie Y firma Z), ścieżka dla niego będzie określona jako Z, Y, X.

Można również spotkać stwierdzenie „certyfikat kwalifikowany” – według ustawy o podpisie elektronicznym²⁹ jest to *„certyfikat spełniający warunki określone w ustawie, wydany przez kwalifikowany podmiot świadczący usługi certyfikacyjne, spełniający warunki określone w ustawie”*.

Certyfikat można tam otrzymać wyłącznie po wykazaniu swojej tożsamości (np. okazaniu odpowiedniego dokumentu). Chodzi oczywiście o to, żeby przykładowo Jan Nowak nie mógł wystąpić o certyfikat i uwierzytelniać się jako Jan Kowalski.

Istnieją również przechowywane i udostępniane przez te podmioty listy certyfikatów unieważnionych (tzw. CRL czyli Certificate revocation list). Może się bowiem zdarzyć, że jakiś klucz prywatny został udostępniony publicznie. Ponieważ nie ma sensu go już dłużej używać, jest dodawany do takiej listy i wszyscy mają obowiązek zaprzestać korzystania z niego.

Warto zauważyć, że tak naprawdę najsłabszym ogniwem w SSL pozostaje człowiek. Mało kto samodzielnie sprawdza elementy takie jak ważność certyfikatu czy porównuje jego dane z informacjami udostępnianymi najczęściej na stronie www podmiotu wykorzystującego dany certyfikat (bezkrytycznie wierząc w brak błędów w odpowiednich procedurach w używanym oprogramowaniu), wiele osób (szczególnie bez wykształcenia informatycznego) ignoruje komunikaty zgłaszane przez programy, gdy te wykryją nieprawidłowości. Wykorzystanie tych spostrzeżeń pozwala na wykonywanie ataków typu „man in the middle”. Polegają one na tym, że dane z komputera użytkownika są kodowane z wykorzystaniem certyfikatu osoby atakującej i przesyłane na jej komputer (gdzie są odszyfrowywane), a dopiero stamtąd (zakodowane właściwym certyfikatem) trafiają do właściwego systemu. Rozwiązaniem mogłoby być wspomniane wcześniej uwierzytelnianie transmisji w obie strony, ale wymagałoby to również np. posiadania certyfikatu i własnych odpowiednio chronionych kluczy przez użytkownika.

2.3. Metody wprowadzania danych identyfikacyjnych do systemów wykorzystujące standardowe urządzenia komputerowe

Pierwsza wyróżniona tutaj grupa metod wymaga od użytkowników posiadania wyłącznie urządzeń, które są obecne w praktycznie każdym współczesnym zestawie komputerowym:

- myszki/rysika, klawiatury
- pamięci masowej

Urządzenia te są - nie trzeba ich dodatkowo instalować i ponosić z tego tytułu żadnych kosztów. Przesądza to o dużej popularności tych rozwiązań.

2.3.1. Wprowadzanie z użyciem myszki/rysika albo klawiatury

Wszystkie metody z tej grupy wykorzystuje się zgodnie ze schematami opisanymi w punktach „Porównywanie z danymi wzorcowymi” i „Wyzwanie-odpowieź”, natomiast nie spotyka się w połączeniu z podpisem elektronicznym. Powód jest prosty – im więcej danych do wpisywania, tym mniej jest to wygodne (może np. występować przy tym więcej błędów) i mniej możliwe do zaakceptowania przez ludzi, którzy mieliby z tego korzystać.

Implementacje metod z tej grupy najczęściej pokazują użytkownikowi, co wpisał (przed wysłaniem do systemu). Zaleca się, aby nie była to dokładna treść. Dlaczego ?

Jest zrozumiałe, że niewskazane jest, aby np. pokazywać na ekranie komputera w biurze dane identyfikacyjne danego użytkownika, można jednak powiedzieć, że nie stanie się nic złego, gdy osoba pracuje pojedynczo w pomieszczeniu (nie ma w nim żadnych innych ludzi) i nie ma tam żadnych kamer, itp. Czy jednak na pewno ?

Dostępne są publikacje (np. Markusa G. Kuhna³⁰) wraz z odpowiednimi przykładami (i publicznymi demonstracjami) na temat zdalnego odczytywania zawartości ekranów CRT/LCD na podstawie generowanego przez nich promieniowania elektromagnetycznego w trakcie ich normalnej pracy (jest to tzw. podsłuch elektromagnetyczny). Publikacje na ten temat pojawiają się również w popularnej prasie komputerowej³¹. Można przypuszczać, że prędzej czy później takie rozwiązania będą powszechnie dostępne na rynku komercyjnym.

Problem ten dotyczy wszystkich metod z grupy „z użyciem klawiatury/myszy”, ale tak naprawdę nie można go zignorować tylko w wypadku metody „hasła wielokrotne” (w przypadku innych w praktyce się to czyni, ponieważ dane tam podawane nie są praktycznie nigdy krytyczne).

Spotyka się rozwiązania, w których system podaje na ekranie znaki zapytania, gwiazdki (zawsze te same znaki), a na miejscu ostatniego znaku to, co zostało rzeczywiście wpisane. Tak zostało zaimplementowane wpisywanie haseł WAP np. w Nokii 6230. Tam może wystarczać dostarczenie użytkownikowi dokładnej informacji, co wpisał i chroniąc to przed osobami trzecimi, w „poważniejszych” rozwiązaniach czegoś takiego jednak nie znajdziemy, gdyż jest to błędne w świetle przedstawionych wcześniej faktów. Lepsze i na szczęście bardziej popularne jest wyświetlanie znaków zapytania, gwiazdek lub innych symboli zamiast wszystkich podawanych znaków (możliwe jest tylko odczytanie z ekranu długości danych). W przypadku długich ciągów może to działać zniechęcająco na osoby trzecie, ale w przypadku krótkich może je dopingować do prób złamania.

Widoczny jest tutaj także drugi problem: jeżeli wykorzystywane jest wpisywanie z klawiatury, to rozwiązanie to całkowicie zawiedzie, gdy w systemie pracuje program lub zainstalowane jest urządzenie przechwytyjące informację o każdym naciśniętym klawiszu (tzw. keylogger). Mogą pozwolić one bowiem osobie atakującej na uzyskanie kompletu informacji niezbędnych do tego, aby działać jako osoba uwierzytelniania. Z tych powodów zaleca się m.in., aby nie podawać swoich danych w trakcie pracy na komputerach dostępnych publicznie – w szkołach, kawiarniach internetowych, itp. Przykładem urządzenia tego typu jest

³⁰ strona domowa autora to <http://www.cl.cam.ac.uk/~mgk25/>

³¹ np. Ireneusz Kubiak, Artur Przybysz, „Zdradziecki prąd”, Chip 1/2005, str. 176 (http://www.chip.pl/arts/archiwum/n/articlear_120593.html)

Key Shark³², wiele programów działających w ten sposób można znaleźć wpisując w przeglądarce google.com hasło „keylogger and download”.

Zagrożenie wynikające z tego można zmniejszać lub wręcz eliminować stosując tzw. klawiatury ekranowe (zwane również wirtualnymi). Rozwiązania tego typu polegają ogólnie na tym, że użytkownik ma wyświetlone na ekranie przyciski z literami (klawiaturę), natomiast „wpisuje” niezbędny tekst klikając myszą w odpowiednich miejscach. Dostępne są programy do różnych systemów operacyjnych pozwalające na wprowadzanie danych w ten sposób do dowolnych innych programów (po „wpisaniu” tekst jest np. przenoszony przez schowek tak, że omija elementy systemu operacyjnego związane z klawiaturą) albo implementacje tej idei wbudowywane bezpośrednio w system, w którym przeprowadzane jest uwierzytelnianie. Przykładem rozwiązania z drugiej grupy (dla aplikacji dostępnych przez przeglądarkę WWW) jest Transec udostępniany przez firmę Macromata na licencji komercyjnej/GPL:

Oprogramowanie po stronie serwera www generuje obrazek zawierający obraz losowo ustawionych „przycisków” i wprowadzonych danych. Jest on pokazywany w przeglądarce użytkownika. Po kliknięciu na obrazek myszką (np. na „przycisku”) do serwera wysyłane są współrzędne punktu, który został kliknięty. Cała interpretacja (czy i co zostało „wpisane”) jest dokonywana po stronie serwera, który zwraca użytkownikowi kolejny obrazek. Możliwe jest również pokazywanie na obrazku „gwiazdek” zamiast wpisanych cyfr i liter. Niestety, pomimo niewątpliwych zalet ta implementacja ma widoczną jedną słabość - jeżeli łącze sieciowe między serwerem i komputerem użytkownika jest wolne lub serwer ma problemy z wydajnością, użytkownik może to odczuć w znaczący sposób.



Rysunek 13. Wygląd przykładowej klawiatury Transec³³

Klawiatury ekranowe podobne do Transec są stosowane np. w bankach internetowych (przykładowo w BZ WBK lub Sez@mie banku BPH). Stosując je należy pamiętać, iż pomimo niewątpliwego utrudnienia dla osób trzecich nie zawsze są one idealne i czasem również można je „złamać” (wystarczy stosowanie programów wykonujących odpowiednio zrzuty ekranu podobnie, jak to się stało w przypadku rozwiązania CityBanku³⁴).

2.3.1.1. Hasła wielokrotne

Hasła są ciągami znaków przypisanymi do użytkowników. Z założenia powinny być znane tylko im (powinni je zapamiętać i nigdzie nie notować). We

³² <http://www.gadgets.co.uk/item/KEYSHARK/Keyboard-Key-Logger.html>

³³ <http://www.micromata.de/produkte/transec.jsp>

³⁴ <http://www.tracingbug.com/index.php/articles/view/23.html>

współczesnych systemach informatycznych spotyka się cztery podejścia do sposobu uwierzytelniania z ich wykorzystaniem:

1. hasło służy do ochrony dostępu do bardziej szczegółowych informacji (które to dopiero w pełni pozwalają zidentyfikować daną osobę)
2. jedno hasło służy do uwierzytelniania jednej operacji (jest to tzw. hasło jednorazowe)
3. hasło jest wykorzystywane wraz z danymi generowanymi losowo w systemie (np. w rozwiązaniach korzystających ze schematu uwierzytelniania „wyzwanie-odpowiedź”)
4. oprócz hasła do każdego użytkownika przypisuje się również drugi ciąg znaków zwany loginem (nazywanym zamiennie nazwą użytkownika)

Metoda z grupy pierwszej jest najczęściej wykorzystywana w połączeniu z przedmiotami przechowującymi dokładne dane o osobie korzystającej z systemu. Podobnie jest z hasłami jednorazowymi. Szerzej tego typu uwierzytelnianie zostało opisane w kolejnych podrozdziałach.

Rozwiązanie wymagające samej klawiatury należy do dwóch ostatnich grup. Przyjmuje się, że użytkownik może być poprawnie uwierzytelniony, gdy poda przypisane do siebie ciągi identyfikacyjne (hasło lub parę login/hasło) bez żadnego błędu.

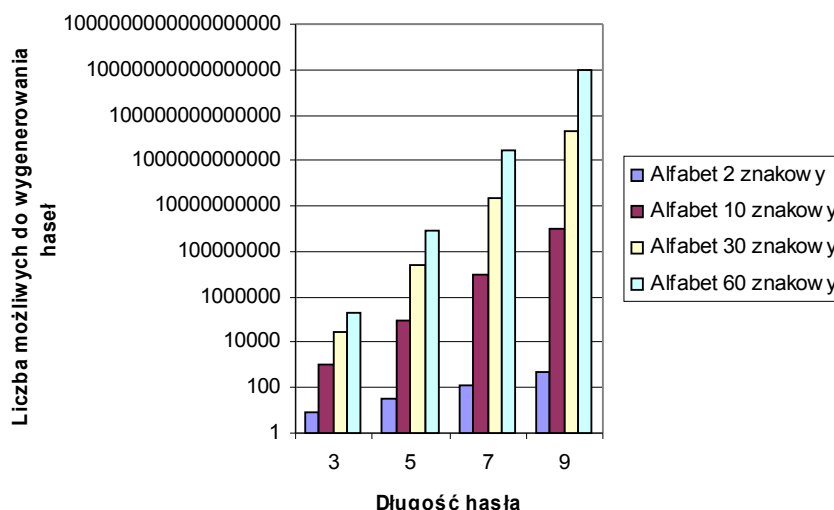
Login bardzo często jest znany nie tylko osobie uwierzytelnianej, ale również innym użytkownikom systemu. Bywa, że jest tworzony według określonych wzorców (np. w firmach może to być nazwisko pracownika, w bankach internetowych jest to zazwyczaj numer klienta będący kolejną liczbą naturalną, itp.). Jest on unikalny, podczas gdy hasła mogą się natomiast powtarzać dla różnych loginów.

Zaleca się, aby hasło było możliwie najdłuższe (przynajmniej kilka znaków) i skomplikowane (zawierać bardzo różne znaki – mieszaninę małych i dużych liter alfabetu, cyfr i innych dostępnych na klawiaturze symboli, które czasem zwane są też znakami specjalnymi). Dlaczego tak się dzieje ?

Ktoś, kto chce uzyskać dostęp do systemu jako upoważniona osoba, może próbować podawać kolejno wygenerowane ciągi identyfikacyjne w nadziei, że w końcu trafi na właściwe (przy czym jak napisano wcześniej, w systemach z loginem tak naprawdę często wystarczy szukać jedynie poprawnego hasła).

Jeżeli włamywacz nie ma informacji, jakie znaki zostały użyte w treści hasła, musi generować ich kolejne możliwe kombinacje dotąd, aż trafi na właściwą (tzw. atak siłowy – „brute force attack”). Im dłużej będą trwać te próby, tym większa szansa na to, że osoba ta zostanie zniechęcona. Warto zauważyć, iż zawsze trzeba się liczyć oczywiście z ryzykiem, że osoba nieupoważniona może „wygenerować” właściwe hasło nawet za pierwszą próbą.

Można w prosty sposób sprawdzić, jak ich liczba i długość hasła wpływają na liczbę możliwych do stworzenia haseł równą ilości prób, które musi wykonać w najgorszym wypadku potencjalny włamywacz (wystarczy podnieść liczbę możliwych do użycia znaków, czyli wielkość alfabetu, do potęgi równej długości hasła). Dane te zostały pokazane na rysunku 14 – widać, iż nie bez powodu w wielu systemach minimalną długość hasła ustawia się na 6 i więcej znaków.



Rysunek 14. Liczba możliwych do wygenerowania haseł w zależności od ich długości i wielkości alfabetu (źródło: opracowanie własne)

Niestety, o ile założenia te są dość dobre, w praktyce ludzie starają się „bronić” przez zbyt skomplikowanymi hasłami. Kilka przykładów takich zachowań:

- zapisywanie haseł obok stanowisk pracy (albo alternatywnie: zapisywanie swoich haseł w różnych programach będącymi rodzajami baz danych lub wręcz w plikach na dysku)
- wykorzystywanie jako haseł wyrazów lub ciągów cyfr powiązanych ze swoim życiem (np. daty urodzenia, imiona żon, inicjałów, itp.)
- używanie jako haseł wyrazów ze słownika (tzw. hasła słownikowe – „dictionary password”)
- wykorzystywanie jednego hasła do uwierzytelniania w wielu systemach

Warto zatrzymać się szczególnie na pierwszym zachowaniu. Jeżeli już takie zapisywanie jest konieczne, ważne jest, aby wykorzystywać do notowania rozwiązania jak najbardziej utrudniające „życie” potencjalnym włamywaczom. Przykładowo:

- hasła powinny być przechowywane w postaci zakodowanej (użytkownik pamiętałby klucz niezbędny do ich odszyfrowania)
- programy przechowywujące tego typu dane po podaniu złego klucza nie pokazywałyby błędów, ale podawały np. wartości losowe

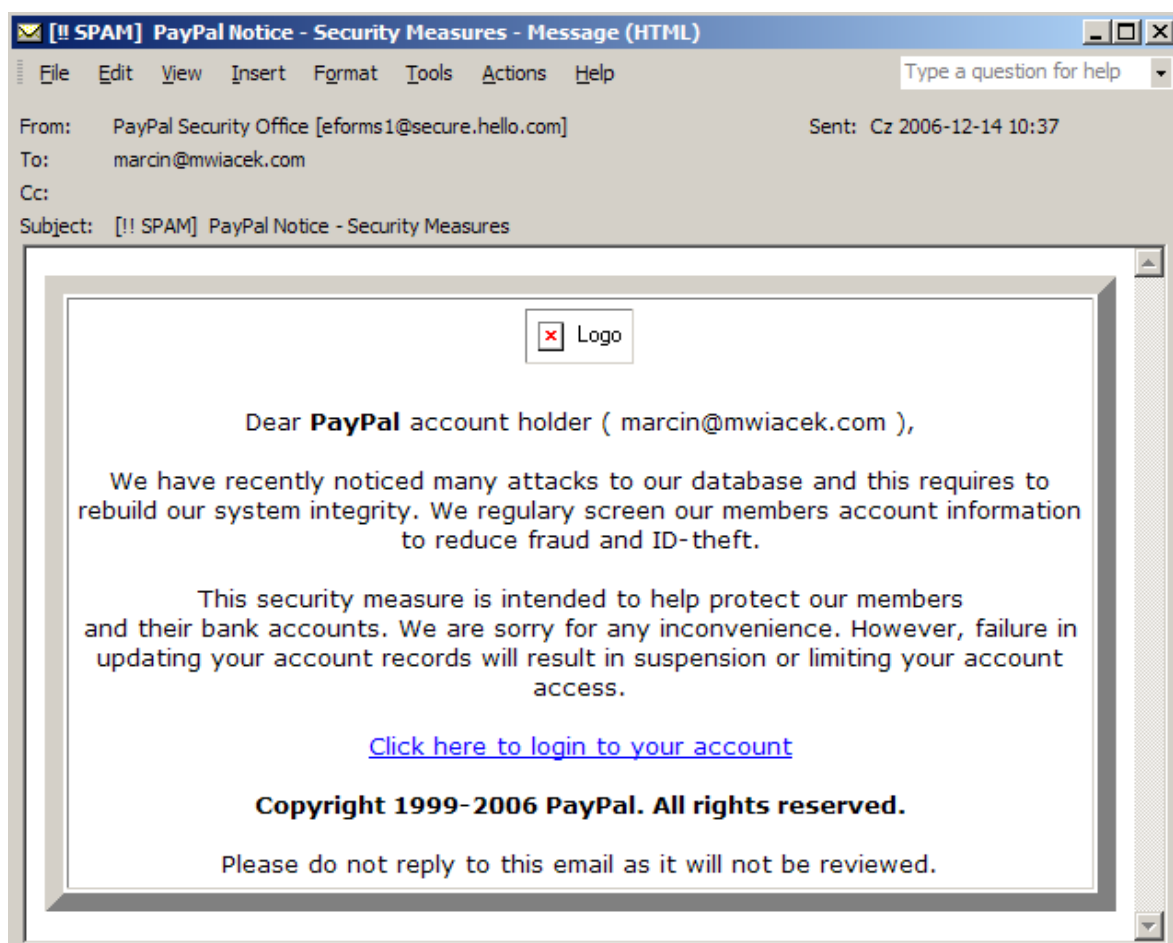
Zdarza się również dobrowolne podawanie haseł przez samych użytkowników po zmanipulowaniu ich w sposób psychologiczny (metody tzw. social-engineeringu).

Są one bardzo różnorodne³⁵, przykładem może być reagowanie na emaile, które wyglądają jak wysłane od rzeczywistych firm i w których klient jest odsyłany na stronę www, na której jest proszony o podanie swoich danych uwierzytelniania

³⁵ dość interesujące przykłady zostały podane w artykule Tomasza Trejderowskiego, „Włamanie do umysłu”, Chip 3/2006, str. 160 (http://www.chip.pl/arts/archiwum/n/articlear_167610.html)

np. w celu odblokowania konta po jego rzekomym wykorzystaniu przez osoby trzecie (strona jest oczywiście fałszywa i służy tylko wyłudzeniu danych – po ich uzyskaniu może np. przekazać je do prawdziwej witryny instytucji, dzięki czemu użytkownik nie ma większych szans na zauważenie swojej pomyłki). Technika ta jest zwana po angielsku phishingiem (od połączenia angielskich słów oznaczających hasło i wędkowanie – password i fishing). Swego czasu dużo tego typu poczty otrzymywali np. użytkownicy banku Inteligo (gdzie uzyskanie nazwy użytkownika i hasła było wystarczające do uzyskania pełnego dostępu do konta).

Część z takiej korespondencji można łatwo wykryć. Przykładem może być email pokazany na rysunku 15 - nie zawiera imienia ani identyfikatora użytkownika, zaś wszystkie adresy (co widać m.in. w jego źródle) prowadzą do innych miejsc niż serwery firmy PayPal. Dodatkowo umieszczony jest w nim odnośnik do logo znajdującego się zewnętrznym serwerze (dostęp do niego został zablokowany w programie pocztowym, co prawdopodobnie zapobiegło wysłaniu osobie wysyłającej potwierdzenia, że email został odebrany i że adres odbiorcy jest poprawny). Z uwagi na obecność tych typowych elementów przesyłek phishingowych wiadomość ta została oznaczona jako spam na komputerze autora. Nie zawsze jednak oprogramowanie jest sobie w stanie poradzić ze współczesnymi próbami phishingu i stąd nie raz jeszcze będzie można usłyszeć o wyłudzeniach tego typu.



Rysunek 15. Przykład emaila służącego wyłudzeniu

Pewnym ułatwieniem dla użytkownika jest domyślna obecność w obecnych przeglądarkach www (np. w Mozilli FireFox od wersji 2.0, Internet Explorerze od wersji 7.0, Operze od wersji 9.1) narzędzi ostrzegających przed wyświetlaniem stron uznawanych za „niebezpieczne”. Podobnie w niektórych pakietach zintegrowanych (np. w Kaspersky Internet Security wywodzącym się z programu antywirusowego Kaspersky Anti-Virus) jeden z modułów próbuje zabezpieczyć użytkownika przed tego typu miejscami. Nie należy jednak całkowicie ufać w te zabezpieczenia.

Można zmniejszać ryzyko nieautoryzowanego dostępu (nie da się go całkowicie wyeliminować):

1. tworząc właśnie bardziej skomplikowane hasła (zgodnie z zasadami podanymi wcześniej). Można używać do tego różne programy do generowania haseł. Trzeba jednak pamiętać, że część z nich tworzy je na podstawie wartości otrzymywanych z generatorów liczb pseudolosowych (a te najczęściej mają powtarzalne wartości). Z tego powodu należy raczej wybierać pakiety, które wykorzystują rzeczywiście losowe liczby – np. powstałe z obserwacji ruchów myszki poruszanej przez użytkownika.
2. stosując nietypowy login (o ile to oczywiście możliwe) – jeżeli np. pracujemy w systemie Windows, nie udostępniamy zasobów jako użytkownik o standardowej nazwie „administrator”
3. wymuszając w systemie blokadę konta danego użytkownika w sytuacji, gdy w przeciągu określonego czasu wystąpiła określona liczba prób uwierzytelniania zakończonych porażką

Ważne jest również, aby sam system był odpowiednio skonstruowany:

1. w przypadku podania złego loginu i/lub hasła nie informował użytkownika, który z nich jest niepoprawny
2. w przypadku podania złego loginu i/lub hasła informował o tym użytkownika zawsze po określonym czasie (w przeciwnym razie przynajmniej teoretycznie w przypadku bardzo wolnego sprzętu po podaniu loginu na podstawie czasu odpowiedzi można np. próbować określić, czy jest on poprawny)
3. nie powinien wyświetlać użytkownikowi na ekranie danych, które ten właśnie wpisał (co było dokładniej wyjaśniane wcześniej)
4. przy zmianie hasła do razu oceniał jego „jakość”
5. nie posiadał tzw. „tylnych drzwi”, czyli nie pozwalał na uzyskanie dostępu w każdych warunkach np. po podaniu określonego przez producenta hasła (co się zdarzało np. w starszych BIOSach do płyt głównych x86³⁶). Widać tutaj, że najlepsze pod tym względem są systemy udostępniane z pełnym kodem źródłowym (np. różne programy typu Open Source), gdyż użytkownik z odpowiednią wiedzą może sam sprawdzić, czy nie ma tam tego typu „niespodzianek”
6. nie stosował haseł „domyślnych” np. w przypadku nowo dodawanych kont (jeżeli użytkownicy nie będą musieli ich zmieniać, to pewna część tego na

³⁶ przedstawione np. w Piotr Metzger, „Anatomia PC” wydanie IX, ISBN 83-7361-507-5, Wydawnictwo Helion, 2004

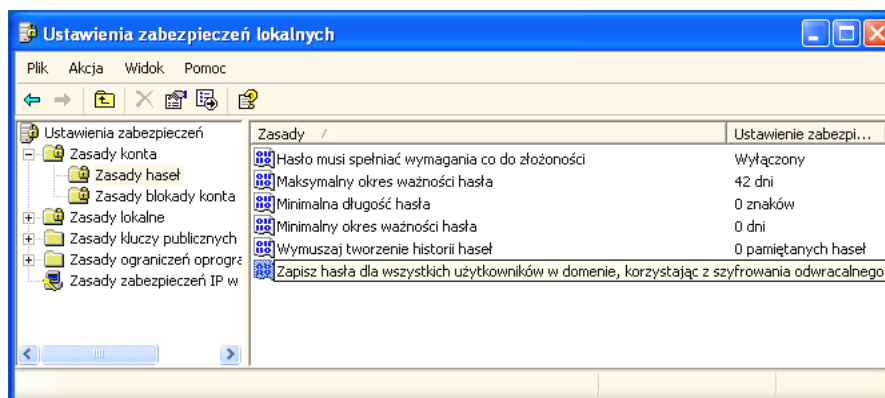
pewno nie robi – na przykładzie raportu firmy Symantec³⁷ dotyczącego urządzeń sieciowych widać, że może to być nawet ich połowa)

Na podstawie powyższych porad można sformułować pewną prawidłowość – im bardziej system korzystający z tego typu metod jest bezpieczny, tym jest mniej „przyjazny” dla użytkowników. Należy tutaj niestety mieć pewne wyczucie w konstruowaniu systemu po to, aby nie doprowadzać do tego, że coraz więcej osób będzie się „bronić” przed zbyt skomplikowanymi hasłami (zgodnie z zasadami podanymi wcześniej).

Jak widać, rozwiązania z hasłami nie są bezpieczne. Mimo to stosujemy je w codziennej pracy narażając się na utratę danych. Przykładem jest konfiguracja wielu serwerów poczty elektronicznej (standardy POP3 i SMTP) – użytkownicy są zmuszeni do wysyłania loginów i haseł otwartym tekstem.

Czy mając to na względzie można jakoś zmniejszyć ryzyko związane z „podśluchaniem” hasła i następnie wykorzystaniem go przez osoby trzecie?

Metoda jest najprostsza z możliwych – hasła należy zmieniać co jakiś czas. W wielu systemach możliwe jest definiowanie, co ile dni albo co ile razy trzeba to robić. Zdarza się, że dostępne jest również przechowywanie historii haseł (po to, aby uniemożliwić zmianę przez użytkownika hasła na jedno z używanych wcześniej). Chociaż nie jest to zbyt wygodne, należy z tych funkcji korzystać (nie można jednak narzucać zbyt drastycznych ograniczeń użytkownikom, ponieważ ci zaczną hasła po prostu zapisywać).



Rysunek 16. Opcje i ich domyślne wartości dla haseł kont użytkowników w Microsoft Windows XP Professional PL (źródło: okno „Narzędzi administracyjnych” w „Panelu Sterowania”)

Na początku tego podrozdziału wspomniano o hasłach wykorzystywanych wraz z danymi generowanymi losowo w systemie. Można tutaj przypisać rozwiązania zastosowane np. w WEP (szyfrowanie zostało opisane w podrozdziale 2.2.2 przedstawiającym schemat uwierzytelniania „wyzwanie-odpowieź”), ale pewną odmianą tego są także wszelkiego rodzaju identyfikatory sesji wykorzystywane w różnego rodzaju aplikacjach i serwisach www. Generowane są one automatycznie (najczęściej jako liczby losowe po stronie serwera) po poprawnym uwierzytelnieniu się użytkownika (podaniu hasła, czasem także loginu) i służą do identyfikacji użytkownika tak długo, dopóki ten działa w systemie. Wszystkie uwagi dotyczące

³⁷ http://www.symantec.com/pl/pl/about/news/release/article.jsp?prid=20070221_01

tego typu haseł (np. o konieczności ich częstej zmiany) pozostają oczywiście aktualne.

2.3.1.2. Odpowiedzi na pytania

Ten rodzaj zabezpieczenia można spotkać głównie w aplikacjach dostępnych poprzez przeglądarkę WWW (jako dodatek do głównej metody uwierzytelniania). Użytkownik przy rejestracji wpisuje treść pytania (albo wybiera ją z kilku proponowanych) i swoją odpowiedź. Jeżeli ma być uwierzytelniany, system wyświetla mu pytanie (ewentualnie prosi o wybór pytania, które było zarejestrowane) i oczekuje podanej wcześniej odpowiedzi. Warto przypomnieć, że zarówno treść pytania jak i odpowiedź nie powinny oczywiście być znane osobom trzecim.

The screenshot shows a web browser window with the address bar displaying 'http://profil.wp.pl/rejestracja.html?idu=99'. The page title is 'Profil - Wirtualna Polska'. The main content area is titled 'Rejestracja' and includes a section for 'Założ nowe konto!'. The form fields are as follows:

- Wybór loginu i hasła:** Login WP to twój pseudonim, którym posługiwać się będziesz korzystając z wszystkich serwisów WP. Do tego dobierz odpowiednie hasło.
- Hasło:** login @wp.pl
- Hasło:** *****
- Powtórz hasło:** *****
- Podpowiedź do hasła:** umożliwi Ci odzyskanie dostępu do konta w sytuacji, gdy zapomnisz hasła. Są na to trzy metody: wysyłając SMS na numer telefonu komórkowego, wpisując prawidłową odpowiedź na pytanie, lub wysyłając nowe hasło na podany przez Ciebie alternatywny adres email.
- Alternatywny email:** [text input]
- Twój telefon komórkowy:** [text input]
- Twoje pytanie:** Wybierz pytanie [dropdown menu]
- Twoja odpowiedź:** [text input]
- Data urodzenia:** [dropdown menu] - [dropdown menu] - [dropdown menu]

The 'Pomoc' section on the right is open, showing a 'Podpowiedź do hasła' section with the text: 'Każdemu może przytrafić się zapomnieć hasła. Ale nie musisz się martwić, jeżeli zapomnisz hasła - pomożemy Ci. Możesz wybrać z listy podpowiedź która zostanie wyświetlona, gdy podasz nieprawidłowe hasło. Nie powinno to być hasło - a jedynie krótki komentarz.'

Rysunek 17. Przykład wykorzystania odpowiedzi na pytania w systemie poczty elektronicznej Wirtualnej Polski (służą wyłącznie do odzyskiwania hasła)

2.3.1.3. Captcha i Hip

Trochę inną funkcję niż hasła (uwierzytelnianie określonego użytkownika) spełniają rozwiązania określane słowem Captcha (według różnych źródeł

internetowych jest to skrót od „Completely Automated Public Turing test to tell Computers and Humans Apart”), spotyka się również skrót Hip (Human Interactive Proof) lub nazwę „turing” (od nazwiska Alana Turinga)³⁸. Ich zadaniem jest bowiem jedynie sprawdzenie, czy użytkownik jest człowiekiem czy nie. Są wykorzystywane w różnych aplikacjach dostępnych poprzez przeglądarki WWW (występują w praktycznie każdym większym serwisie) i służą do blokowania możliwości edycji danych występujących w tych samych formularzach różnego rodzaju automatów.

Captcha to najczęściej automatycznie generowany obrazek zawierający grafikę połączoną ze znakami (cyframi i/lub literami). Użytkownik musi prawidłowo odczytać i podać te znaki. Z założenia są one pisane różnymi krojami pisma i kolorami, przekracane, cieniowane, itp., a grafika jest tak na nie nakładana, aby nie można było zastosować OCR (Optical Character Recognition), czyli programów i funkcji do automatycznego rozpoznawania pisma.



Rysunek 18. Przykład obrazka typu captcha generowanego w formularzu do tworzenia konta poczty elektronicznej w portalu Gazeta.pl

W praktyce część prostych implementacji obrazków captcha jest tak skonstruowana, że ostatnia własność bywa niespełniona. Istnieje zresztą szereg stron internetowych opisujących słabości różnych rozwiązań (jedną z ich list można otrzymując np. wpisując w przeglądarce google.com hasło „breaking and captcha”). Kolejnym problemem jest to, że skrypty generujące kolejne obrazki zawierają błędy i zdarza się, że tworzą powtarzalne wzory.

Your edit includes new URL links; as a protection against automated spam, you'll need to type in the words that appear in this image:
(What is this?)

83 - 4 =

Rysunek 19. Rozwiązanie typu captcha zaproponowane w rozszerzeniu ConfirmEdit do pakietu MediaWiki³⁹

Warto zauważyć również, że tego typu rozwiązania mogą utrudniać dostęp do niektórych systemów osobom z ograniczeniami (np. słabo widzącym). Z tych względów próbuje się np. łączyć obrazki wraz z hasłami jednorazowymi, zamiast obrazków prosić użytkownika o wpisanie wyniku działania matematycznego (jak na rysunku obok), rysować litery trójwymiarowo (rozwiązanie tEABAG_3D ze strony OCR Research Team) albo wręcz „odczytywać” głosowo użytkownikowi to, co ma wpisać (wysyłając mu plik dźwiękowy z tekstem do odtworzenia – jest to

³⁸ formularz dodawania użytkownika w <http://www.moneybookers.com>

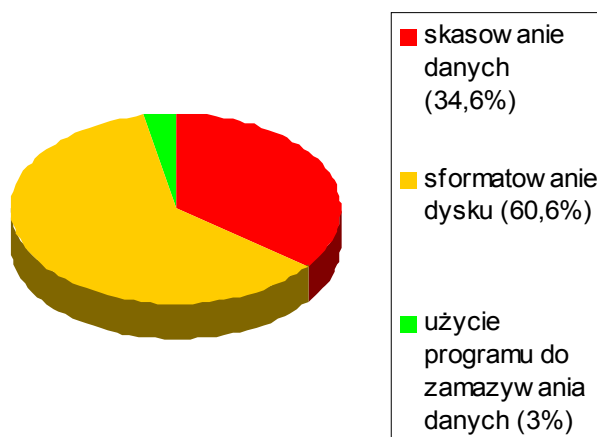
³⁹ <http://www.mediawiki.org>

dostępne np. w systemie poczty elektronicznej Gmail). Proponuje się również wyświetlanie zdjęć kotów i psów – użytkownik musi wybrać te pierwsze. Przykładem implementacji tego ostatniego pomysłu może być projekt Asirra (Animal Species Image Recognition for Restricting Access)⁴⁰ stworzony przez Microsoft Research.

2.3.2. Wprowadzanie z plików

Ta grupa metod wykorzystuje kolejną grupę standardowych urządzeń występujących w systemach informatycznych, jaką są wszelkiego rodzaju urządzenia do przechowywania plików – dyski twarde, dyskietki, karty flash (MMC, SD, itp.), pamięci flash podłączane do USB, itp. System przy uwierzytelnianiu wymaga dostępu do zapisanych tam plików. Spotyka się tutaj rozwiązania związane z podpisem elektronicznym (proponuje się zapisywać na nośnikach wymiennych klucze prywatne użytkowników), ale również z hasłami.

Problemem jest możliwość skopiowania zawartości nośnika (bez pozostawienia widocznych śladów). Z tego powodu przedmiot z danymi powinien być chroniony fizycznie przez użytkownika przed osobami trzecimi, jak również (zgodnie z zasadą „nie można wykraść czegoś, co jest odłączone”) dane z niego powinny być dostępne w systemie komputerowym tylko wtedy, gdy jest to rzeczywiście niezbędne. Dlatego też (ale także ze względu na sposób działania podpisu elektronicznego) pewnym nieporozumieniem jest udostępnianie możliwości przechowywania (przez cały czas) prywatnych kluczy użytkowników (służących do autoryzowania operacji) na serwerach banku, jak to się dzieje np. w rozwiązaniu Sez@m banku BPH. I nie zmienia tego fakt, iż są tam chronione hasłem.



Rysunek 20. Dane z eksperymentu opisanego w magazynie „Chip”⁴¹ pokazujące, jak użytkownicy usuwają dane z dysków twardych

Inne zagrożenie wynika z tego, że wielu użytkowników komputerów nie kasuje skutecznie danych z nośników. Jeżeli zostaną one sprzedane (i znajdą się na nich dane identyfikacyjne pierwotnego właściciela), nabywca może bardzo często z nich

⁴⁰ <http://research.microsoft.com/asirra/>

⁴¹ Szymon Piłat, Kamil Kulesza, „Niebezpieczne pozostałości”, Chip 7/2006, str. 140 (http://www.chip.pl/arts/archiwum/n/articlear_174400.html)

skorzystać. Odpowiednie badania pokazujące skalę tego zjawiska były już wykonywane dla dysków twardych – w eksperymencie magazynu Chip jedynie 3% użytkowników usunęło swoje informacje poprawnie (kolor zielony na rysunku 20), podobne próby w krajach o wyższej świadomości informatycznej⁴² dały niewiele lepsze rezultaty (9%). Rozwiązaniem jest np. edukacja użytkowników, częsta zmiana danych identyfikacyjnych w używanych systemach i/lub szyfrowanie ich na nośniku.

Można zalecić również, aby rzeczywiste dane niezbędne do przeprowadzenia uwierzytelniania były „ukrywane” w plikach zapisanych na nośniku (np. w zbiorach z grafiką z użyciem steganografii) albo żeby nośnik był używany w sposób niestandardowy (przykładowo – dyskietka z danymi była tak spreparowana, że próba odczytu z poziomu Windows nie ujawnia na niej żadnych plików, a program wykorzystujący zawarte na niej dane po prostu czyta je bezpośrednio z odpowiednich sektorów). Nie wyeliminuje to oczywiście problemu, pozwoli jednak przynajmniej w jakimś stopniu na zmniejszenie ryzyka, że osoby trzecie po wykradzeniu takiego przedmiotu (albo po odnalezieniu go, gdy właściciel go zwyczajnie zgubi) nie będą potrafiły go wykorzystać (dodatkowo: jeżeli „prawdziwy” użytkownik zauważy odpowiednio szybko fakt utraty nośnika, będzie miał szansę na zmianę swoich danych identyfikacyjnych w systemie). Jest to ważne tym bardziej, że często wykorzystywane są coraz mniejsze nośniki (np. karty pamięci flash), które łatwo zgubić.

Zdarza się, że wprowadzanie danych z plików stosuje się jedynie pomocniczo. Rozwiązanie tego typu zastosowano np. w technologii kodowania dysków Bitlocker dostępnej w Microsoft Windows Vista – użytkownik może umieścić klucz kodujący w module TPM (przedstawiony w podrozdziale 2.4.7), zapisać go na nośniku USB albo nawet wprowadzać z klawiatury (co wpływa na utratę funkcjonalności i bezpieczeństwa).

2.4. Metody wprowadzania danych identyfikacyjnych do systemów wymagające posiadania specjalnych przedmiotów

Opisane tutaj zostanie grupa metod, w których użytkownik musi posiadać przy sobie odpowiedni przedmiot. Zapisane są na nim w sposób „tradycyjny” dane, które trzeba wprowadzić do systemu (przykładowo: należy wpisać z klawiatury to, co podano na kartce papieru). W innych rozwiązaniach przedmiot zawiera dane w formie elektronicznej. Po włożeniu go lub przyłożeniu do odpowiedniego czytnika, system weryfikuje tożsamość użytkownika na podstawie danych odczytanych z czytnika.

Wady tego typu metod są oczywiste:

1. przedmiot można zapomnieć zabrać ze sobą (co za tym idzie, bez niego być niedopuszczonym do systemu pomimo posiadania odpowiednich uprawnień)
2. jeżeli przedmiot zostanie zgubiony lub odebrany z użyciem siły, mogą z niego skorzystać osoby trzecie (o ile oczywiście nie zastosowano dodatkowych zabezpieczeń)

⁴² Simson L. Garfinkel i Abhi Shelat „Remembrance of Data Passed: A Study of Disk Sanitization Practices”, MIT, styczeń 2003 (http://www.computer.org/portal/cms_docs_security/security/v1n1/garfinkel.pdf)

Wady te można wyeliminować wbudowując przedmiot w ciało użytkownika (np. wszczepiając go pod skórę). Możliwości techniczne istnieją, najbardziej chyba znane próby z tym związane przeprowadził prof. Kevin Warwick z Wydziału Cybernetyki Uniwersytetu Reading w Wielkiej Brytanii⁴³, jest możliwe, że tego typu rozwiązania są już w użyciu. Należy jednak zauważyć, że nie można stosować tego na masową skalę (choćby z uwagi na obawy ludzi przed utratą prywatności) oraz, że w skrajnych sytuacjach (gdyby stało się to powszechnością) może dojść do wypadków wyjmowania przedmiotu siłą z ciała użytkownika.

Inny problem związany jest z tym, czy dany przedmiot faktycznie chroni zapisaną na nim informację przed osobami trzecimi. W wielu wypadkach można bowiem tylko wierzyć zapewnieniom producenta i mieć nadzieję, że nawet osoby z odpowiednim sprzętem nie będą w stanie sforsować zabezpieczeń (jeżeli bowiem im to by się udało, ze zrozumiałych względów mogłyby nie być zainteresowane publicznym rozgłaszaniem tego faktu i użytkownicy mogliby być nieświadomi). Można zaproponować przykład takiej sytuacji: jeżeli dysk twardy zabezpieczony jest sprzętowo hasłem, dane nie powinny być dostępne po wymianie płytki z elektroniką. Z tych powodów warto wybierać raczej przedmioty skonstruowane według pewnych standardów (co z drugiej może obniżyć bezpieczeństwo np. z uwagi na dostępność czytników) niż rozwiązania własne firm.

Metody tego rodzaju są używane zarówno z podpisem elektronicznym (oczywiście tylko, jeżeli oczywiście przedmiot ma wystarczającą pojemność na zapisanie odpowiednich danych takich jak np. klucze prywatne użytkownika), ale także ze wszystkimi innymi wymienionymi wcześniej schematami uwierzytelniania.

Grupa tych metod będzie się niewątpliwie powiększać. Już teraz można sądzić, że w przyszłości prawdopodobnie pojawią się systemy, w których użytkownik będzie uwierzytelniany na podstawie zdjęcia wykonanego jego aparatem cyfrowym (np. w telefonie komórkowym). Okazuje się bowiem⁴⁴, że tego z uwagi na błędy matryc w tego typu konstrukcjach możliwe jest ich praktycznie jednoznaczne odróżnianie.

2.4.1. Numery seryjne programów

Producenci oprogramowania bardzo często zapisują np. w plikach na płytach instalacyjnych programów numery seryjne (tzw. klucze). Numery te znajdują się również na dołączanych plakietkach albo w dołączonych książkach. Programy podczas instalacji porównują to, co zostało zapisane na płytach z tym, co zostanie wpisane przez użytkownika. Jeżeli dane są dokładnie takie same, program przyjmuje, że został zainstalowany przez swojego nabywcę.

Można się oczywiście zastanawiać, czy jest sens wymieniać numery seryjne jako metodę uwierzytelniania z co najmniej kilku powodów:

1. numery te nie są często bowiem żadną tajemnicą (w wielu wypadkach istnieją narzędzia do ich odzyskiwania⁴⁵ albo jeżeli program jest w wersji OEM, to najczęściej plakietka z tymi danymi jest na obudowie komputera)

⁴³ jego strona domowa to <http://www.kevinwarwick.com/>

⁴⁴ Jan Lukáš, Jessica Fridrich i Miroslav Goljan, „*Digital Camera Identification from Sensor Pattern Noise*”, czerwiec 2006 (<http://www.ws.binghamton.edu/fridrich/>)

⁴⁵ przykładowo: w przypadku wielu wersji Windows po ich zainstalowaniu można użyć jednego z wielu programów, który poda numer seryjny po jego uruchomieniu w działającym systemie

2. nie są one unikalne (zdarza się, że te same numery seryjne są umieszczane w wielu kopiach pakietów oprogramowania)
3. często są generowane z użyciem określonego algorytmu (a nie losowe)
4. pewne wątpliwości powstają, gdy zaczniemy się zastanawiać, czy proces ten to rzeczywiście tylko uwierzytelnianie czy również autoryzacja

Zostały one wymienione w tym miejscu, ponieważ mają wiele cech haseł wielokrotnych i wymagają przedmiotu (jakim jest plakietka lub książeczka), a ich celem jest „sprawdzenie” tożsamości użytkownika.

Poza tym mają jednak cechy uwierzytelniania, a nie autoryzacji – w ich przypadku (w przeciwieństwie do kluczy licencyjnych) raczej rzadko zdarza się, że zawierają np. datę wygaśnięcia (co mogłoby pozwolić zaklasyfikować proces jako sprawdzenie służące do ustalenia, czy użytkownik ma w dalszym ciągu prawo do używania produktu). Ten podział jest zresztą znacznie bardzo wyraźny w przypadku produktów z aktywacją (np. Microsoft Windows XP). Po zainstalowaniu z użyciem numeru seryjnego (ustaleniu tożsamości nabywcy) programy te wymagają bowiem wyraźnej autoryzacji (w celu sprawdzenia, czy nabywca nie zainstalował np. zbyt dużej ilości kopii) za pomocą aktywacji.

2.4.2. Hasła jednorazowe

W jednym z poprzednich podrozdziałów (2.3.1.1) przedstawione zostało wykorzystywanie haseł wraz z sugestią, żeby je zmieniać co jakiś czas. Można pójść dalej i hasło zmieniać po jednym użyciu (będą to wtedy tzw. hasła jednorazowe). Przy takim podejściu login może być oczywiście niepotrzebny, ale w praktyce tak nie jest. Użytkownik raczej nie ma szans za zapamiętanie każdego możliwego hasła i stąd zawsze otrzymuje ich wykaz (ewentualnie musi mieć przy sobie swój telefon GSM). Ta koncepcja jest stosowana powszechnie np. przez różne banki internetowe:

- w mBanku posiadacz konta otrzymuje zwykłą pocztą kartkę z ponumerowanymi zestawów cyfr, system przy każdej operacji prosi o zestaw cyfr o kolejnym (i to jest słabość) numerze. Alternatywnie hasło może być wysłane do klienta SMSem na jego numer telefonu komórkowego.
- w Inteligo do klienta wysyłana jest plastikowa karta zawierająca wydrukowane hasła ułożone podobnie jak w mBanku. Różnica jest w tym, że użytkownik musi zdrapywać kolejne hasła. Jak można sądzić, rozwiązanie to przez to zyskuje (jeżeli na hasle jest zdrapka, możliwe jest, że nikt go nie użył) ale i traci (jeżeli lista haseł zostanie przechwycona np. w wyniku kradzieży, to wiadomo będzie, które hasło będzie użyte jako następne).
- kolejne hasła są generowane za pomocą urządzenia zwanego tokenem (posiada ono wyświetlacz, na którym są one wyświetlane). Czasami zabezpieczony jest on dodatkowo własnym kodem PIN. Istnieją również tokeny, które generują hasła cały czas zmieniając je np. co minutę (w ich wypadku pewnym problemem jest konieczność synchronizacji zegarów tokena i odpowiednich elementów po stronie systemu potwierdzających poprawność hasła tak, aby tworzone przez nie wartości były rzeczywiście takie same).



Rysunek 21. Token LUKAS Banku (źródło: strona [www LUKAS Banku](http://www.lukasbanku.pl)⁴⁶)

Jak widać, niektóre rozwiązania z tej grupy są lepiej zabezpieczone przed podejrzeniem kolejnych haseł niż inne. Wyraźnie trzeba jednak zaznaczyć, iż nawet najlepiej zaimplementowane hasła jednorazowe nie zagwarantują całkowitego bezpieczeństwa. Jeżeli zostaną one bowiem przechwycone w drodze między użytkownikiem i systemem, intruz będzie mógł je wykorzystać (dokładnie jeden raz) do swoich celów. Warto przypomnieć o problemie „keyloggerów” – stąd można zalecić wprowadzanie haseł jednorazowych np. przez klawiatury ekranowe.

2.4.3. Karty stykowe z paskiem magnetycznym

Karta wykonana jest w postaci prostokątnej plastikowej płytki z naniesionym paskiem magnetycznym, gdzie zapisane są dane elektroniczne. Oprócz tego na karcie najczęściej nadrukowane lub wytłoczone są różne informacje identyfikacyjne (dzięki nim w określonych sytuacjach w razie uszkodzenia paska możliwe jest uwierzytelnienie użytkownika karty przez osoby trzecie). Spotyka się więc: numer karty i datę ważności, imię i nazwisko właściciela, czasem także jego zdjęcie i odręczny podpis.

Kartę (dokładniej: pasek magnetyczny) trzeba włożyć do czytnika albo przesunąć z odpowiednią prędkością bezpośrednio przy głowicy czytającej (stąd określenie „karta stykowa”). Czasami może to być uciążliwe dla użytkownika.

Użycie tego rozwiązania wiąże się też z pewnymi kosztami (karty, czytniki, infrastruktura). Aby je obniżyć, wiele implementacji wykorzystuje najpopularniejsze dostępne na rynku standardy (ponieważ istnieje wiele zgodnych z nimi urządzeń i rozwiązań, ich producenci konkurują ze sobą m.in. obniżaniem cen). Najczęściej można mówić o spełnianiu normy ISO/IEC 7810 (określa rozmiar kart – tam jako ID-1 jest zdefiniowana najpopularniejsza chyba obecnie na rynku wielkość: wysokość 53,98 mm x szerokość 85,60 mm x grubość 0,76 mm), rodziny norm ISO/IEC 7811 (definiują m.in. rozmieszczenie maksymalnie trzech ścieżek z danymi) czy też normy ISO/IEC 7813 (podaje dokładniejsze wymagania dotyczące kart płatniczych). Niestety, popularność ma też swoją drugą cenę - łatwiejsze jest wykorzystanie jednego z dostępnych na rynku czytników i wykonanie duplikatu karty. Z uwagi na to same dane elektroniczne nie powinny być wystarczające do przeprowadzenia procesu uwierzytelniania.

O tej zasadzie zapomniano zaś np. w przypadku kart płatniczych. Z tego to powodu tak często zaleca się ich użytkownikom, aby nie tracili ich z oczu (i dlatego obecnie w praktycznie żadnym miejscu nie powinno zabierać się karty na zaplecze, ale przynosić terminal do klienta).

⁴⁶ http://www.lukasbank.pl/oferta_ekonto_bezpiecz_token.asp

Można oczywiście powiedzieć, że część urządzeń (np. bankomaty) przy próbie wykonania operacji z użyciem karty tego typu wymaga podania kodu PIN (ciągu 4 cyfr). Trzykrotne złe podanie tego numeru spowoduje zatrzymanie karty i konieczność kontaktu z instytucją, która ją wydała. Niestety nie daje to jednak całkowitego bezpieczeństwa. Zdarzało się już, że w niektórych bankomatach grupy przestępcze instalowały urządzenie do skanowania paska magnetycznego (w miejscu, gdzie karta była wkładana do urządzenia) i miniaturową kamerę (dzięki której można było odczytać kod PIN). Uzyskanie tych danych było wystarczające do przygotowaniu działającego duplikatu karty. O tego typu kradzieżach można przeczytać od czasu do czasu w komunikatach Policji.

Jak widać, zagrożenie może się kryć również tam, gdzie użytkownik się go najmniej spodziewa. Pojawiło się nawet pojęcie „skimmingu”, czyli wykorzystania zawartości paska magnetycznego bez wiedzy właściciela karty.

Ponownie trzeba też niestety wspomnieć o „phishingu” – okazuje się, że w przypadku przynajmniej niektórych kart płatniczych podanie nadrukowanych na nich danych jest też całkowicie wystarczające do przeprowadzenia transakcji. Powoduje to, że różne grupy przestępcze tworzą maile i witryny www, gdzie użytkownik w imię weryfikacji czy odblokowania jakiegoś swojego konta jest proszony o podanie szczegółów swojej karty kredytowej (m.in. numeru i daty ważności). Kończy się to oczywiście kradzieżą pieniędzy.



Rysunek 22. Przykład karty płatniczej z paskiem magnetycznym (źródło: galeria wypukłych kart płatniczych serwisu KartyOnline⁴⁷)

W przypadku istniejących systemów wprowadza się wprawdzie różne ulepszenia, ale tak naprawdę nie eliminują one prawdziwych przyczyn problemów związanych z tą technologią.

I tak proponuje się wydawanie oddzielnych kart do transakcji elektronicznych (np. eKARTA z mBanku) albo udostępnia za darmo (przykładowo dzieje się tak w mBanku dla kart debetowych Visa Electron wydawanych dla rachunku eKONTO, eMAX i izzyKONTO⁴⁸) funkcję ustawienia maksymalnego limitu (najczęściej dziennego i miesięcznego) transakcji dla karty zalecając ich zmniejszenie do niezbędnego minimum. I choć użytkownik przy ewentualnym wyłudzeniu może stracić mniejszą kwotę, ryzyko tego faktu jest takie samo.

Udostępnia się kod CVV2 (zwany również CVC2 lub CID), który nie jest zapisywany na pasku. Z założenia ma być on wykorzystywany do autoryzowania transakcji elektronicznych, w praktyce jednak nie wszystkie sklepy go wymagają.

⁴⁷ <http://www.kartyonline.pl/gal1.php>

⁴⁸ zgodnie z cennikiem <http://www.mbank.com.pl/przewodnik/oplaty-pelne-1.html> w dniu 27 grudnia 2006

Jak widać rozwiązanie to, choć popularne, ma wiele wad i zdążyło się już źle zapisać w świadomości wielu użytkowników. Z wyżej wymienionych powodów odchodzi się od niego.

2.4.4. Karty stykowe z układami elektronicznymi

W tym rozwiązaniu wbudowywuje się układy elektroniczne w plastikową płaską obudowę (na zewnątrz wyprowadzona jest zazwyczaj jedynie jakaś forma interfejsu połączeniowego – np. powierzchnie stykowe). Układy najczęściej zasilane są z zewnątrz (z czytnika). Zdarza się, że na powierzchni są umieszczane dane właściciela (imię, nazwisko, itp.) podobnie jak w przypadku kart stykowych z paskami magnetycznymi. Całość zwana jest kartami inteligentnymi (smart cards) czy też kartami ze zintegrowanymi układami (Integrated Circuit(s) Cards czyli ICC), a potocznie kartami chipowymi.

Podobnie jak w przypadku kart z paskiem magnetycznym najczęściej zachowuje się zgodność z normami ISO/IEC (m.in. ze standardem ISO/IEC 7810 oraz rodziną ISO/IEC 7816). Również tutaj użycie karty wymaga włożenia jej do czytnika (przy czym z uwagi na inną budowę wystarczy, że znajdzie się tam część ze stykami). Stosowane są dwa podejścia:

- w karcie umieszczone są wyłącznie układy pamięciowe (rozwiązanie tańsze)
- element zawiera pamięć i mikroprocesor, który udostępnia zawartość pamięci wyłącznie po podaniu odpowiedniego kodu i czasami spełnia także różne inne funkcje (np. szyfrowania sprzętowego)



Rysunek 23. Karta hybrydowa Visa Electron wydana przez Bank Zachodni WBK S.A. (źródło: galeria płaskich kart płatniczych serwisu KartyOnline⁴⁹)

Pierwsza możliwość (obecność wyłącznie układów pamięciowych) z punktu widzenia procesów uwierzytelniania jest mało interesująca – przedmiot taki może pełnić co najwyżej rolę pamięci masowej.

Z kolei układ z mikroprocesorem, jak podano wcześniej, może spełniać różne funkcje. Od zabezpieczenia zawartości kodem (zazwyczaj po kilkakrotnym złym podaniu kodu karta jest bezpowrotnie blokowana lub wymaga podania kodu odblokowującego), zliczania impulsów (np. w kartach telefonicznych) po „bezpieczne” generowanie, przechowywanie i operowanie (prywatnymi) kluczami użytkownika w standardach takich jak RSA. Urządzenie z tymi ostatnimi możliwościami zwane jest również kartą kryptograficzną lub kryptoprocessorową.

⁴⁹ <http://www.kartyonline.pl/gal2.php>

Najczęściej spełnia np. jeden ze standardów z rodziny PKCS (Public Key Cryptography Standards) firmy RSA Data Security, Inc. i może być np. wykorzystane do przeprowadzania procesu logowania w systemie Windows. Przykładem jest CryptoCard multiSIGN firmy CryptoTech. W sektorze bankowym wydawane są również tzw. karty hybrydowe (z układami elektronicznymi, ale także z paskiem magnetycznym). Można się także spotkać z informacjami o kartach Java. Tak się składa, że część z tych produktów jest tak zaawansowana, że posiada nawet własny system operacyjny. Jest on napisany tam właśnie w tym języku programowania. Jak różnorodna jest oferta rozwiązań, można zobaczyć na stronie jednej z produkujących je firmy (np. Gemplus).

Przykładem karty mikroprocesorowej mogą być karty SIM (Subscriber Identification Module) stosowane w telefonach GSM czy karty USIM stosowane w telefonach UMTS. Jeżeli nie zapisano na nich inaczej, po włączeniu telefonu należy podać kod, aby możliwe było odczytanie danych identyfikacyjnych numeru użytkownika i zalogowanie się do sieci komórkowej. Wtedy też udostępniana jest książka telefoniczna i wiadomości SMS (w niektórych rozwiązaniach, jak np. starszych kartach prepaid SimPlus, wymagany jest dodatkowo dodatni stan konta). Można przypuszczać, że w przyszłości stosowane będą nie tylko do przechowywania danych użytkownika lub związanych z jego identyfikacją w sieci, ale również informacji niezbędnych do składania podpisu elektronicznego z użyciem telefonu (mowa jest o tzw. mobilnym podpisie elektronicznym).



Rysunek 24. Karta SIM (źródło: serwis Jakuba Bakxa⁵¹)

Należy dodać, że karty stykowe z układami elektronicznymi nie są oczywiście całkowicie bezpieczne. Istnieje szereg metod pozwalających na dostęp do danych z obejściem zabezpieczeń procesora⁵⁰:

1. techniki mikroanalizy (microprobing) polegające na bezpośredniej obserwacji i ingerencji w strukturę układów (są one w przeciwieństwie do innych inwazyjne, gdyż niszczą powierzchnię karty i układu). Najczęściej wykorzystywane są tutaj takie elementy jak np. mikroskop elektronowy. Możliwe jest np. podłączanie się do punktów testowych układów (używanych normalnie w fabryce do

⁵⁰ na podstawie Oliver Kömmerling i Markus G. Kuhn „Design Principles for Tamper-Resistant Smartcard Processors”, 10-11 maj 1999 (<http://www.cl.cam.ac.uk/~mgk25/>)

⁵¹ http://www.bakx.pl/index.php?id=museum/karty_sim/karty_sim

- sprawdzenia poprawności wykonania) czy bezpośrednie obejrzenie zawartości komórek pamięci.
2. ataki programowe wykorzystujące słabości w konkretnych implementacjach (przykładowo: jak to opisano na witrynach przedstawiających projekt SIMemu⁵², możliwe jest np. klonowanie starszych kart SIM, gdyż nie były one zabezpieczane przez nieskończoną ilością prób odczytania kluczy dostępowych)
 3. obserwacja i analiza promieniowania elektromagnetycznego wydzielanego przez mikroprocesor w trakcie działania (zmienia się ono w trakcie wykonywania różnych czynności)
 4. generowanie kontrolowanych błędów poprzez próby dostępu niezgodne ze specyfikacją karty (np. podawanie na styki zbyt wysokich napięć)

Szczególnie trzy ostatnie są niebezpieczne, gdyż w skrajnym wypadku właściciel karty może nawet nie zauważyć, że odczytano z niej zabezpieczone dane. Możliwe jest również oczywiście przygotowanie takiego terminala, który oprócz spełniania swoich podstawowych funkcji będzie również przekazywać dane z karty włamywaczowi.

Mimo to należy stwierdzić, że rozwiązanie to ze względu na swój sposób działania może spełniać swoją rolę w procesach uwierzytelniania znacznie lepiej niż karty z paskiem magnetycznym.

2.4.5. Karty i przedmioty zbliżeniowe

Tym razem dane przechowywane w przedmiocie są przekazywane do czytnika w momencie zbliżenia przedmiotu do czytnika (dlatego mówi się czasem również o rozwiązaniach bezstykowych). Przedmioty bywają różne (mogą mieć wielkość kart kredytowych, ale również wszelkiego rodzaju żetonów), zdarza się, że pełnią również rolę kart chipowych lub z paskiem magnetycznym. Podobnie jak w przypadku innych kart stosuje się również umieszczanie na nich danych właściciela – tak jak np. w przypadku przepustek samochodowych w Wojskowej Akademii Technicznej. Tego rodzaju przedmioty można wykorzystać do każdego rodzaju uwierzytelniania – wszystko zależy m.in. od tego, ile danych można zapisać w ich pamięci.



Rysunek 25. Wzór chipowej Elektronicznej Legitymacji Studenckiej (źródło: załącznik nr 3 do Rozporządzenia Ministra Edukacji Narodowej i Sportu z dnia 18 lipca 2005 w sprawie dokumentacji przebiegu studiów⁵³)

⁵² np. <http://www.simemu.pl/>

⁵³ http://www.men.gov.pl/prawo/wszystkie/rozp_361.php

Najczęściej dane są przekazywane radiowo, chociaż ciągle spotyka się także rozwiązania wykorzystujące podczerwień (tam jednak problemem jest konieczność skierowania nadajnika w przedmiocie bezpośrednio w czujnik czytnika). Część przedmiotów zawiera własne zasilanie (nazywane są aktywnymi i mają wewnątrz baterię, którą trzeba raz na jakiś czas wymienić), czasami również przycisk (dopiero po jego naciśnięciu możliwe jest odczytanie danych), w użyciu są też rozwiązania, gdzie zasilanie jest dostarczane przez pole elektromagnetyczne wytwarzane przez czytnik (tzw. pasywne).

W tym drugim przypadku pojawia się problem polegający na możliwości wzbudzenia karty przez osoby trzecie i próby odczytania zawartości karty nie tylko przy czytniku połączonym z systemem. Jest on szczególnie widoczny, gdy stosuje się technologię w rodzaju RFID (Radio Frequency IDentification) pozwalającą na działanie w odległości kilku m od czytnika. Można zmniejszyć prawdopodobieństwo takiego zdarzenia tak tworząc kartę, żeby wymagała bezpośredniej bliskości wzbudzającego jej urządzenia albo wyeliminować je nosząc ją w odpowiednim ekranującym opakowaniu (oczywiście kosztem wygody).

Warto zauważyć, że za rozwiązanie tego typu można w jakimś stopniu uważać karty lojalnościowe wydawane przez sieci handlowe (np. Skarbonka⁵⁴ sieci Auchan). Są one wielkości karty kredytowej, ale działają na zupełnie innej zasadzie. Zawierają bowiem na sobie nadrukowaną kodem paskowym (najczęściej jakąś odmianą kodu EAN - European Article Number) informację, która jest skanowana i odczytywana diodami LED lub laserem w kasie (na jej podstawie dokonywane jest uwierzytelnianie klienta).

Liczba znaków możliwych do zapisania na takiej karcie jest znacznie ograniczona i rozwiązanie to może się nadawać co najwyżej do zapisania jakiegoś identyfikatora lub hasła. W przypadku typowych kodów kreskowych dodatkowym problemem jest fakt, iż zawartość kodu jest zapisywana otwarcie obok niego (wystarczy więc np. sfotografować kartę, aby móc odtworzyć to, co jest na niej zakodowane).

Rozwiązaniem w przypadku małej liczby danych może okazać się stosowanie dwuwymiarowych (2D matrix codes) kodów kreskowych (gdzie dane nie są kodowane w kolejnych paskach, a w matrycy punktów wewnątrz prostokąta lub okręgu). Przykładem mogą być standardy VeriCode i VSCode firmy Veritec Inc., QR Code, BeeTagg, QuickMark czy ShotCode. Problemem ciągle pozostaje możliwość wykonania zdjęcia. Istnieją już zresztą aplikacje nawet do telefonów komórkowych, które potrafią analizować i odczytywać teksty zapisane z użyciem przedstawionych rozwiązań.



Rysunek 26. Tekst "Marcin Wiacek" zapisany w standardzie QuickMark⁵⁵

⁵⁴ <http://www.auchan.pl/karta-skarbonka.html>

⁵⁵ wygenerowany bezpłatnie na stronie <http://www.quickmark.com.tw>

Kolejne firmy proponują też trójwymiarowe kody kreskowe (3D matrix codes), które pozwalają na zapisanie jeszcze większych ilości danych (nawet rzędu kilkaset kB). Być może tam nie będzie możliwości zrobienia zdjęcia (skopiowania danych bez fizycznego naruszenia stanu przedmiotu). Ciągłe jednak pozostaje kwestia wygody (karta musi być umieszczona w polu widzenia czytnika w trakcie odczytu) i potrzeba dbania o stan fizyczny tego przedmiotu (kodu nie można zdrapać i uszkodzić).

2.4.6. Tokeny

Nazwą token określono już wcześniej przedmiot wydawany np. użytkownikom banków i służący do generowania haseł jednorazowych.

Można się również spotkać z innym rodzajem tokenów. Są to bowiem także urządzenia z modułem kryptograficznym (podobnym jak w przypadku kart stykowych z układami elektronicznymi i służącym np. do generowania i przechowywania kluczy prywatnych użytkownika) podłączane do systemu komputera z użyciem interfejsu USB.

Przykładem może być eToken PRO firmy Aladdin. Spełnia on standardy Crypto API i PKCS#11 i może być wykorzystany np. do uwierzytelniania użytkownika w systemie Windows.

2.4.7. TPM

Nazwa TPM (Trusted Platform Module) jest używana w odniesieniu do dwóch przedmiotów:

1. specyfikacji opracowanej przez Trusted Computing Group
2. układu scalonego wyprodukowanego przez jedną z kilku firm (m.in. Infineon) zawierającego fizyczną implementację (niektórych) elementów podanych w tej specyfikacji

W dalszej części pracy nazwa ta będzie używana w odniesieniu do wspomnianego układu. Z założenia przyjmuje się, że wyniki przez niego zwracane są wiarygodne. Potrafi on wykonywać operacje związane przynajmniej z algorytmami RSA (tworzenie kluczy, kodowanie, dekodowanie) i SHA-1 (wykonywanie skrótów) oraz generować liczby losowe. Zawiera własną pamięć przechowującą dane bez zasilania używaną m.in. do przechowywania kluczy szyfrowych.

Układy takie są umieszczane w płytach głównych standardu x86 (sprzedawanych do zestawów stacjonarnych albo notebooków). Najczęściej są montowane tam na stałe i spełniają wymagania poziomu 2 opisanego w dokumencie FIPS 140-2 (czyli w praktyce próba fizycznego sprzętowego dostępu do danych zapisanych w pamięci układu musi być związana z pozostawieniem na nim widocznych śladów). Czasem układ można łatwo wyjąć i zamienić (np. w płytach głównych P5B-Plus firmy ASUSTek Computer Inc., gdzie jest umieszczany w dodatkowej karcie rozszerzającej). To ostatnie rozwiązanie spotyka się jednak dosyć rzadko i z tego względu można stwierdzić, że TPM to po prostu urządzenia z modułem kryptograficznym przypisane do konkretnego systemu komputerowego

(producenci udostępniają zresztą odpowiednie sterowniki, dzięki którym jest uzyskana zgodność np. z jednym ze standardów PKCS).

Jak widać, uwierzytelnianie związane z tą technologią wymagałoby wprowadzenia odpowiednich danych identyfikacyjnych do pamięci układu TPM (który musiałby być włączony) w każdym systemie, z którego miałby korzystać użytkownik. Z tego powodu rozwiązanie może być przydatne co najwyżej pomocniczo w pewnych specyficznych przypadkach. Przykładem może być wykorzystanie tego modułu w notebookach firmy Apple z procesorem x86 (Intel Core Duo), gdzie dane tam zawarte umożliwiają uruchomienie systemu operacyjnego Mac OS X.

Warto dodać, że jest to rozwiązanie kontrowersyjne. Jest bowiem częścią większej całości zwanej Trusted Computing pozwalającej uzyskać (przynajmniej w teorii) m.in. możliwość pełnej kontroli nad tym, jakie programy i pliki są otwierane i uruchamiane w konkretnym systemie⁵⁶. Specyfikacja udostępniana przez Trusted Computing Group zakłada wprawdzie, że moduł TPM można wyłączyć, jednakże prawdopodobnie nie wszędzie (np. we wspomnianych komputerach Apple) jest to możliwe.

2.4.8. Klucze sprzętowe

Programy komputerowe można również zabezpieczać korzystając z tzw. kluczy sprzętowych (dongles). Program po uruchomieniu szuka takiego klucza i z jego pomocą sprawdza poprawność licencji użytkownika (przy czym może to być zarówno uwierzytelnianie jak i autoryzacja). Jeżeli któraś z tych czynności zakończy się niepowodzeniem, wyłącza się.

Klucze takie mogą być podłączane do portu drukarkowego (popularnego kiedyś Centronics), obecnie spotyka się również rozwiązania działające z USB, w formie kart Compact Flash czy też PC Card (dawniej nazywane PCMCIA). Każde takie urządzenie zawiera własną pamięć, gdzie zapisywane są dane identyfikacyjne użytkownika (czasami także np. licznik ilości uruchomień programu). Zdarza się, że udostępnia również sprzętowy generator liczb losowych

Rozwiązanie to stosuje raczej tylko do zabezpieczania specjalistycznego oprogramowania (programy serwisowe czy inżynierskie) z uwagi na jego koszt implementacji. Można zaryzykować stwierdzenie, że zastępuje się je standardowymi modułami kryptograficznymi (np. kartami stykowymi czy też tokenami).

Przykładem kluczy sprzętowych są produkty z serii HASP firmy Aladdin czy DESKey firmy Data Encryption Systems.

2.5. Metody wprowadzania danych identyfikacyjnych do systemów związane z cechami użytkowników

Tym razem przyjęto inne założenie – danymi przyjętymi do uwierzytelniania (z wykorzystaniem schematu „porównywanie z danymi wzorcowymi”) będą dane stworzone na podstawie cech fizycznych samych użytkowników systemu. Można

⁵⁶ warto przeczytać tekst Petera Gutmanna o możliwych skutkach kontroli tego typu zastosowanej w Microsoft Windows Vista - oryginał http://www.cs.auckland.ac.nz/~pgut001/pubs/vista_cost.txt lub polski przekład http://byte.livenet.pl/?page_id=819

stwierdzić, że w idealnych warunkach ta grupa powinna dawać najlepsze rezultaty – nie trzeba przecież nic pamiętać ani nic posiadać, wystarczy tylko się znaleźć w odpowiednim miejscu.

Metody te można podzielić na wykorzystujące sprawdzanie jednej cechy (metody biometryczne) albo wielu cech (sprawdzanie tożsamości użytkownika przez innego człowieka).

2.5.1. Metody biometryczne

Metody biometryczne bazują na sprawdzaniu jednej cechy (fizjologicznej, czyli związanej z cechą elementu ciała albo behawioralnej, czyli związanej ze sposobem zachowania) u osoby uwierzytelnianej i wymagają umieszczenia odpowiednich czytników w każdym systemie, w którym ma być uwierzytelniany użytkownik. Cecha musi być oczywiście tak wybrana, żeby można było ją zmierzyć i żeby metoda spełniała w jak największym stopniu przedstawione wcześniej zasady uwierzytelniania:

1. liczba błędnych reakcji systemu (niepoprawne uwierzytelnienie osoby upoważnionej i poprawne intruza) musi być jak najmniejsza – w tym konkretnym przypadku można to osiągnąć wykorzystując cechę:
 - występującą u wszystkich użytkowników
 - inną u różnych osób w mierzalnym stopniu (umiemy więc sprawdzać np. wzrost, ale jest on nieprzydatny, ponieważ jesteśmy w stanie stwierdzić, że jest wiele osób o tej samej wysokości)
 - niezależną od stanu zdrowotnego, zmęczenia czy też nastroju
 - niemożliwą do skopiowania (taką, żeby przygotowanie wzorca i oszukanie nim czytnika było niewykonalne lub zbyt kosztowne)
 - niezmienną w czasie (cechą nie może być np. kolor ust)
 - niezależną od warunków otoczenia (np. wilgotności powietrza) i którą można zmierzyć niezależnie od ich zmiany
2. system powinien być możliwie przyjazny dla użytkownika (ten musi przede wszystkim akceptować m.in. sposób odczytu – przykładowo prawdopodobnie niewykonalne było zmuszenie osób uwierzytelnianych do pobierania krwi)
3. rozwiązanie ma być jak najtańsze

Są to oczywiście wymagania idealne, do których się dąży, ale które trudno spełnić. W praktyce określa się teoretyczną liczbę błędnych reakcji systemu na maksymalnie określonym poziomie i wykorzystuje cechy, które umożliwią zrealizowanie tego założenia (np. jeżeli ma być badana jakaś cecha głosu użytkowników, na podstawie m.in. badań statystycznych określa się, czy akceptowalne jest, że określona liczba osób w założonym przedziale czasu może chorować w sposób zmieniający barwę ich głosu na tyle, że system nie będzie ich poprawnie uwierzytelniał).

Zdarza się, że zasada druga i trzecia jest traktowana marginalnie – w określonych zastosowaniach (np. strategicznych) najważniejszy jest fakt, iż niektóre cechy fizyczne jest bardzo trudno podrobić i system taki warto stosować (będzie bowiem znacznie bardziej „pewny” niż rozwiązania tradycyjne) bez względu na koszty czy wygodę.

Jest to również wada – ponieważ wzorzec biometryczny z założenia jest niezmienny, po udostępnieniu jego cyfrowej reprezentacji (np. w wyniku błędu w systemie, kradzieży danych czy wręcz ich sprzedaży) użytkownik może być szpiegowany wszędzie tam, gdzie będą dostępne czytniki związane z tym wzorcem (i nie będzie miał możliwości obrony przed tym).

Dosyć sugestywnie (właśnie w sensie negatywnym) zostało to pokazane w filmie „Raport mniejszości”⁵⁷ – czytniki danych z oczu były praktycznie w każdym miejscu publicznym i gdziekolwiek bohater się pojawił, można było go zlokalizować.

Warto zauważyć, że czytniki powinny być odporne na próby oszukania ich przez podłożenie martwych części ciała. Niestety problemem ciągle pozostaje fakt, że użytkownik może być doprowadzony do czytnika siłą...

Niezależnie od tego - w miarę rozwoju technologii na pewno znajdowane będą kolejne metody tego typu. W literaturze można przeczytać informacje o próbach i pierwszych rozwiązaniach związanych z analizą:

1. dynamiki pisania na klawiaturze (odstęp między naciskaniem i puszczeniem klawiszy) - np. BioPassword firmy BioPassword, Inc.⁵⁸
2. DNA (materiał można pobierać chociażby ze śliny) – np. w firmie Biowell
3. chodu (sposobu poruszania się)
4. ruchu warg (może to być połączone z analizą głosu) – np. w firmie DCS
5. zapachu – można znaleźć informacji o „elektronicznym nosie” Cyranose 320⁵⁹
6. termogramu twarzy
7. struktury opuszka palca (ang. Fingertip Recognition) – technologia PosID polegająca na tworzeniu obrazu wewnętrznej struktury opuszka palca
8. struktury łoża paznokcia (ang. Nail Bed Recognition)
9. pojemności elektrycznej paznokcia i fragmentu palca – firma FnBiometrics proponuje przyklejanie w określonych miejscach nadajników RFID, które mogą przekazywać radiowo wspomniane (charakterystyczne dla osobnika) dane. Tego typu podejście (stosowanie rozwiązania radiowego) niesie oczywiście ze sobą dodatkowe zagrożenia, które opisano w podrozdziale „Karty i przedmioty zbliżeniowe”
10. struktury skóry – np. rozwiązanie firmy Lumidigm, w którym rejestruje się obraz utworzony przez światło o różnej długości fali po jego odbiciu się od powierzchni fragmentu powierzchni ciała

2.5.1.1. Odciski linii papilarnych

Każda osoba posiada na swoich palcach listewki skórne zwane liniami papilarnymi (epidermal ridges). Dotykając różnych przedmiotów pozostawiamy na ich powierzchni odciski mniejszych lub większych fragmentów tych linii.

Badanie odcisków palców w celu identyfikacji osób zostało powszechnie zastosowane na długo przed wprowadzaniem do użytku systemów informatycznych (dziedzina zajmująca się tym to tzw. daktyloskopia, a za pierwszą pozycję zawierającą pełną klasyfikację linii papilarnych przyjmuje się wydaną w

⁵⁷ „Minority Report” Stevena Spielberga z Tomem Cruisem (rok 2002, wytwórnie 20th Century Fox i Dreamworks Pictures) nakręcony na podstawie opowiadania Philipa K. Dicka „The Minority Report”

⁵⁸ <http://www.biopassword.com>

⁵⁹ <http://trace.smithsdetection.com/products/Default.asp?Product=60§ion=Military>

1892 „*Finger prints*” Francisa Galtona⁶⁰). Charakteryzują się one bowiem następującymi cechami⁶¹:

1. niepowtarzalnością (są one inne nawet dla bliźniaków jednojajowych)
2. nieusuwalnością (jak podaje artykuł „*zmiany w wyglądzie linii papilarnych mogą być spowodowane tylko i wyłącznie przez głębokie rany, oparzenia lub choroby organizmu*”)
3. niezmiennością

i pozwalają na wydanie jednego z trzech stwierdzeń:

- „Ślad linii papilarnych pochodzi od osoby
- Ślad linii papilarnych nie pochodzi od osoby
- Ślad linii papilarnych nie nadaje się do badań identyfikacyjnych.”,

zaś samo porównywanie śladu linii papilarnych z wzorcem polega na tym, że „Wykonujący badania bierze pod uwagę występujące wzory linii, tzw. minucje, czyli cechy charakterystyczne budowy listewek skórnych oraz odległości między nimi.” (jest tylko jedno zastrzeżenie: „Zakwalifikowany do badań ślad powinien cechować się przynajmniej minimalną liczbą cech charakterystycznych.”).

Istnieje kilka rodzajów czytników linii papilarnych:

- optyczne
- pojemnościowe
- naciskowe
- termiczne
- ultradźwiękowe

Urządzenia należące do pierwszej grupy to rodzaj skanerów optycznych, które wykonują cyfrowe zdjęcie przyłożonego palca. Nowsze konstrukcje wykorzystują matryce CCD (palec jest przykładany do lusterka, na lusterko z kolei kierowane jest promieniowanie, które to po odbiciu od palca jest kierowane na matrycę). Wadą tego rozwiązania jest wrażliwość na wszelkiego rodzaju brud i zabrudzenia oraz brak odporności na fałszerstwo (istnieje możliwość poprawnego rozpoznania nawet zwykłego zdjęcia).



Rysunek 27. Przykład czytnika linii papilarnych⁶²

⁶⁰ <http://galton.org/books/finger-prints/index.htm>

⁶¹ wg. artykułu „Dwanaście cech linii” Komendy Stołecznej Policji w Warszawie (http://www.ksp.waw.pl/laboratorium/download/Publikacje/Dwanascie_cech_linii.pdf)

⁶² zdjęcie ze sklepu internetowego (adres http://www.tajne.eu/sklep/product_info.php?products_id=606)

Skanery pojemnościowe z kolei wykorzystują matryce kondensatorów. Każdy z nich zwiększa swoją pojemność wraz ze zbliżaniem do nich skóry w danym miejscu. Wartości są następnie zamieniane przy przetworniki analogowo-cyfrowe. Przykładem skanerów tego typu są urządzenia firmy Authentec obecne m.in. w notebookach firmy HP (np. czytnik AES2501 dołączony przez USB w modelu nx6325), Toshiba, Fujitsu Siemens, itd.

Urządzenia naciskowe zawierają matryce czujników nacisku (tam, gdzie na palcu znajduje się linia, styka się z sensorem, który rejestruje nacisk).

Skanery termiczne wykorzystują fakt różnicy temperatur między powietrzem i powierzchnią palca (przy czym nie ma tutaj mowy o konkretnych temperaturach, ale bardziej o różnicy między tymi dwoma elementami). Tego typu czytniki produkuje np. firma Atmel (rozwiązanie FingerChip⁶³).

Z kolei wzmiankę o czytniku ultradźwiękowym można znaleźć w materiałach firmy Optel⁶⁴. Zasada działania jest następująca: *„Na powierzchnię, do której przytknięty jest analizowany obiekt kierowana jest od strony prawej fala dźwiękowa. Sygnały rozproszone kontaktowo przez obiekt odbierane są przez przetwornik, wykonujący ruch po kole o osi prostopadłej do powierzchni kontaktu.”*⁶⁴. Wydaje się, że technologia ta nie wyszła poza fazę prototypu.

Warto dodać, że niezależnie od rodzaju czytnika proponuje się użytkownikom na wszelki wypadek zapisywanie w systemie odcisków kilku palców.

2.5.1.2. Rozpoznawanie kształtu twarzy

W metodzie tej wykorzystywane jest zdjęcie, na podstawie którego następnie system uwierzytelnia pozytywnie lub odrzuca użytkownika. Problemem okazuje się uzyskanie poprawnego zdjęcia – wynik może zafałszować np. używanie nakrycia głowy albo jej obrócenie o pewien kąt. Pierwsze systemy wymagały dodatkowo, aby było ono wykonywane w ściśle określonych warunkach – przykładowo użytkownik musiał być fotografowany na tle o określonym kolorze. Stosowane jest kilka podejść:

- użycie metod bazujących na transformacie Karhunen-Loevego zwanej również analizą składowej głównej (PCA - Principal Component Analysis). Wykorzystywane są tam metody statystyczne – z ich pomocą *„eliminowane są elementy silnie skorelowane, prowadząc do zestawu słabo zależnych liczb tworzących wektor cech twarzy”*⁶⁵.
- poszukiwanie na zdjęciach charakterystycznych elementów twarzy i badanie odległości między nimi. Przykładem może być metoda dopasowania grafu (EGM - Elastic Graph Matching)⁶⁶ czy też metoda zaproponowana przez Rein-Lien Hsu⁶⁷

⁶³ <http://www.atmel.com/products/Biometrics/>

⁶⁴ <http://www.optel.com.pl/article/polska/art.htm>

⁶⁵ Adam Czajka, Andrzej Pacut, „Twój PIN to TY, część II” (Biuletyn NASK, marzec-kwiecień 2003, str. 18-24, 2003)

⁶⁶ Martin Lades, Jan C. Vorbrüggen, Joachim Buhmann, Jörg Lange, Christoph v.d. Malsburg, Rolf P. Würtz i Wolfgang Konen, „Distortion Invariant Object Recognition in the Dynamic Link Architecture”, IEEE transactions on computers, vol. 42, no. 3, marzec 1993 (<http://www.vision.caltech.edu/CNS179/papers/Lades93.pdf>)

⁶⁷ Rein-Lien Hsu, Mohamed Abdel-Mottaleb i Anil K. Jain, „Face detection in color images” (http://www.cg.cs.uni-bonn.de/docs/teaching/2002/WS/cv_hand_tracking/documents/papers/face-detection-in-color.pdf)

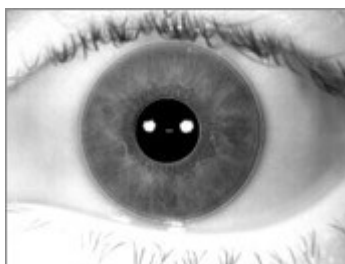
- wykorzystywanie efektu tzw. „czerwonych oczu”⁶⁸.

W zaproponowanych algorytmach oprócz metod statystycznych wykorzystuje się również elementy takie jak sieci neuronowe⁶⁹. Pomimo wielu propozycji nie udało się uzyskać całkowitej rozpoznawalności. Powstało już wiele systemów komercyjnych: np. produkty firmy Viisage takie jak FacePASS czy FaceFINDER.

2.5.1.3. Badanie tęczówki oka

W metodzie tej badany jest obraz tęczówki oka (ang. iris). Zmienia ona wprawdzie swój kolor wraz z wiekiem, jednak jej wzór jest praktycznie stały przez całe życie. Warto od razu zaznaczyć, że to ostatnie stwierdzenie stoi w sprzeczności z tzw. irydologią, która zakłada, że w tęczówce widoczna jest informacja o chorobach człowieka. Bezspornym natomiast problemem jest oczywiście konieczność eliminacji z odczytanego przez czytnik obrazu m.in. rzęs, powiek, itp.

Dziedziną tą zajmuje się szeroko m.in. John G. Daugman z Uniwersytetu w Cambridge (wiele informacji na ten temat można znaleźć na jego stronie domowej⁷⁰). Jest on m.in. właścicielem patentu amerykańskiego numer 5 291 560 „*Biometric personal identification system based on iris analysis*”. Na nim opierają się rozwiązania Iridian Technologies⁷¹.



Rysunek 28. Przykład zdjęcia tęczówki oka⁷²

Istnieje również kilka innych rozwiązań związanych z tą metodą. Przykładowo firma IriTech Inc.⁷³ używa metody zgodnej z patentem USA nr 6 247 813.

Rozwiązania związane z tęczówką zostały użyte do potwierdzenia tożsamości tzw. „afgańskiej dziewczynki” w 2002 (przez porównanie zdjęcia osoby z okładki magazynu National Geographic z 1985 ze zdjęciem zrobionym kilka lat później⁷⁴), wykorzystuje się ich w bankomatach, na lotniskach (np. lotnisko Narita w Japonii),

⁶⁸ Carlos Morimoto, Dave Koons, Arnon Amir i Myron Flickner, „*Real-Time Detection of Eyes and Faces*”, IBM Research Center, USA, 1998 (<http://www.acm.org/icmi/1998/Papers/Morimoto.pdf>)

⁶⁹ np. Henry A. Rowley, Shumeet Baluja i Takeo Kanade, „*Neural NetworkBased Face Detection*”, School of Computer Science, Carnegie Mellon University, Pittsburgh, 1996 (<http://imagelab.ing.unimo.it/tte/varie/rowley96neural.pdf>)

⁷⁰ <http://www.cl.cam.ac.uk/~jgd1000/>

⁷¹ <http://www.iridiantech.com>

⁷² zdjęcie z materiałów informacyjnych o Pracowni Biometrii NASK/Politechniki Warszawskiej dostępnych pod adresem <http://www.nask.pl/run/n/PracowniaBIO>. Według nich nierozwiązanym problemem jest ciągle „kontrola żywotności oka podczas dokonywania pomiaru”.

⁷³ <http://www.irittech.com>

⁷⁴ <http://www.cl.cam.ac.uk/~jgd1000/afghan.html>

w szpitalach (np. szpitalu miejskim w Bad Reichenhall w Niemczech), na granicach czy w różnych koncernach.



Rysunek 29. Czytnik tęczówki oka BM-ET330 firmy Panasonic⁷⁵

2.5.1.4. Rozpoznawanie wzoru żył

W metodzie wykorzystuje się fakt, że kształt naczyń krwionośnych pozostaje stały przez całe życie (zmieniają się jedynie ich rozmiary). Użytkownik musi przyłożyć do czytnika określoną część ciała (np. konkretny palec). System rejestruje obraz utworzony przez promieniowanie podczerwone (długość fali około 760 nm) po jego przejściu przez badany fragment ciała – w miejscach, gdzie płynie krew w żyłach, widoczne są ciemne obszary (promienie są bowiem pochłaniane przez hemoglobinę), w pozostałych obszarach obraz pozostaje jasny.

Metoda jest bardzo interesująca z punktu widzenia biometrii, ponieważ pomiar jest nieszkodliwy i nie wpływają na niego takie czynniki jak zanieczyszczenia powierzchniowe sprawdzanego fragmentu ciała (naczynia krwionośne umieszczone są przecież głęboko pod skórą). Brak jest tutaj bariery psychologicznej (przykładowo – odciski palców czy rozpoznawanie kształtu twarzy na podstawie zdjęć kojarzone są często z policją, co skutecznie może zniechęcać do ich stosowania wiele ludzi).

Technologia ta jest szeroko stosowana np. przez Fujitsu (nazywana jest tam Palm Vein⁷⁶). Firma ta podawała w lutym 2005, że współczynnik fałszywego odrzucenia dla jej czytników wynosi 0,01%, a współczynnik fałszywej akceptacji mniej niż 0,00008%. Według niektórych źródeł⁷⁷ używa się ich już w ponad 15 000 miejsc, zostały natomiast zastosowane m.in.:

- w bankomatach japońskiego banku Tokyo-Mitsubishi
- w szpitalu uniwersytetu w Tokio (do zabezpieczenia dostępu do pomieszczeń serwerowych i zapisów medycznych)
- w firmie Meiwa Estate Company
- w bibliotece miejskiej w mieście Naka w Japonii

⁷⁵ zdjęcie z

<http://catalog2.panasonic.com/webapp/wcs/stores/servlet/ModelDetail?displayTab=O&storeId=11201&catalogId=13051&itemId=88595&catGroupId=21552&surfModel=BM-ET330>

⁷⁶ <http://www.fujitsu.com/global/about/rd/200506palm-vein.html>

⁷⁷ <http://www.itportal.com/absolutenm/templates/article-biometrics.aspx?articleid=3566&zoneid=50>



Rysunek 30. Przykład obrazu żył w dłoni⁷⁸

2.5.1.5. Badanie kształtu dłoni

Użytkownik musi położyć dłoń na czytniku. Mierzone są dokładne wymiary m.in. palców (długości i ich szerokości) i samej dłoni np. po oświetleniu jej światłem podczerwonym. Odpowiednie położenie dłoni wymuszone jest zazwyczaj ogranicznikami. Problemem w przypadku tej metody okazuje się np. możliwość oszukania czytnika przez podstawienie odpowiedniej protezy (stąd musi być ona stosowana pomocniczo – wraz z innymi rozwiązaniami).

Przykładem może być czytnik HandKey II⁷⁹ firmy Schlage Recognition Systems. Rozwiązania tego typu były i są wykorzystywane np. w trakcie olimpiady w 1996 w Atlancie, w bankach (Eastern Financial Florida Credit Union) czy w szpitalach (Aspirus Wausau Hospital w Wausau, Wisconsin).

2.5.1.6. Badanie podpisu odręcznego

Pismo odręczne (a dokładniej jego kształt) jest ciągle uważane za element odmienny u różnych ludzi. Z tego powodu często stosuje się sprawdzanie autentyczności dokumentów tworzonych ręcznie przez porównywanie ich z innymi fragmentami, co do których autorstwa nie ma wątpliwości (wykonywana jest wtedy tzw. analiza grafologiczna). Badane są cechy takie jak kształt i wielkość czy odstępy między poszczególnymi literami.

W systemach komputerowych użytkownik musi napisać np. na odpowiedniej tabliczce rysikiem pewien tekst. Dokonywana jest jego analiza i porównanie z cechami wzorców. Możliwe jest natomiast badanie również samego sposobu pisania (np. szybkości czy nacisku pisaka), co dodatkowo zwiększa skuteczność działania (wygląd pisma można bowiem skopiować, ale przyjmuje się, że sposób pisania już nie). Rozwiązanie tego typu (tablet z pisakiem) zostało skonstruowane np. w Laboratorium Biometrii NASK i Politechniki Warszawskiej. Można również kupić Biometric Pen (Bio-Pen) firmy Secure Signature Systems⁸⁰. Wygląda on jak

⁷⁸ zdjęcie z <http://www.european-hospital.com/topics/article/826.html>

⁷⁹ <http://recognitionssystems.schlage.com/products/product.php?id=2>

⁸⁰ <http://www.securesignaturesystems.com>

zwykły długopis, współdziała z platformą x86 (połączenie przez USB), zaś według producenta FAR wynosi 0,01%.

Innym ciekawym produktem związanym z pismem odręcznym jest system SignHear firmy Sign Assured Ltd.⁸¹. Użytkownik musi tam złożyć swój podpis, a urządzenie bada dźwięk, jaki powstaje w trakcie tej czynności.

2.5.1.7. Rozpoznawanie głosu

Uwierzytelnianie jest dokonywane na podstawie wypowiedzianego do mikrofonu określonego (zawsze tego samego) lub dowolnego podanego przez system tekstu (ta druga możliwość eliminuje możliwość przyjęcia nagrania). Problemem okazują się zmiany głosu wynikające z czynników takich jak zmęczenie lub choroba oraz szумы z otoczenia. Z tych powodów przy konstruowaniu systemów biometrycznych producenci starają się raczej wykorzystywać cechy związane z samym procesem wytwarzania dźwięku przez użytkownika (wynikające ze sposobu mówienia oraz różnic w kształcie i wielkości narządów mowy).

Przykładem rozwiązania może być SpikeServer⁸² firmy Diaphonic Inc. Jest to platforma, którą można zastosować np. w banku. Pozwala ona bowiem na uwierzytelnianie ludzi dzwoniących telefonicznie.

2.5.1.8. Badanie siatkówki oka

Czytniki tego typu rejestrują obraz wzoru naczyń krwionośnych znajdujących się tuż pod siatkówką (ang. retina) oka po oświetleniu ich światłem podczerwonym.

Wadą tego rozwiązania jest koszt, brak wygody (konieczność zdjęcia okularów, szkieł kontaktowych, itp. oraz ustawienia się osoby sprawdzanej w określonej pozycji względem czytnika), jak również przynajmniej teoretyczna możliwość uszkodzenia naczyń w wyniku oświetlenia ich wysyłanym światłem podczerwonym.



Rysunek 31. ICAM 2001⁸³

Przykładem może być ICAM 2001⁸⁴ firmy Rayco Security. Jak przyznaje sam producent, błąd odrzucenia uprawnionych użytkowników (FRR) jest duży (0,1%), a bardzo istotnym problemem jest konieczność akceptacji skanowania oka przez użytkowników. Równocześnie jednak w materiałach dotyczących tego urządzenia

⁸¹ <http://www.signassured.com/>

⁸² <http://www.diaphonics.com/spikeproduct.php>

⁸³ zdjęcie z <http://www.raycosecurity.com/biometrics/EyeDentify.html>

⁸⁴ <http://www.raycosecurity.com/biometrics/EyeDentify.html>

jest informacja, że współczynnik fałszywej akceptacji (FAR) jest równy 0,0001% (można też spotkać stwierdzenie, że „*brak jest przypadków fałszywej akceptacji*”), a całość nadaje się stosowania „*w systemach kontrolnych o wysokich standardach bezpieczeństwa takich jak reaktory atomowe lub instalacje militarne*”.

2.5.1.9. Identyfikacja związana z uchem

Metody odróżniania ludzi na podstawie kształtu ich ucha były i są przedmiotem prac amerykańskiego szeryfa Alfreda Lannarelli. Zgodnie z jego metodą zdjęcie prawego ucha powinno być odpowiednio powiększone (chodzi o wyeliminowanie różnic wynikających np. z innych odległości badanych ludzi od czytnika), a następnie szuka się na nim określonych punktów (ważne jest szczególnie odnalezienie pierwszego z nich) i dokonuje pomiarów odległości między nimi.

Oprócz analizy wykonywanej na podstawie fotografii wykorzystuje się również wykonywanie obrazu termalnego ucha. Pozwala to na łatwe wyeliminowanie takich problemów jak zasłonięcie ucha np. włosami (mają one bowiem niższą temperaturę).

Identyfikacja oparta na uszach opiera się również na pobieraniu znaczników po przyścisnięciu ucha do powierzchni czytnika albo wykonywaniu pomiaru akustycznego.

2.5.2. Uwierzytelnianie przez innego człowieka

W systemach wojskowych lub o znaczeniu strategicznym użytkownicy są sprawdzani i dopuszczani do systemu przez żywą osobę. Jest to praktycznie całkowicie bezpieczne rozwiązanie (człowiek jest w stanie porównywać bardzo dużo cech na raz), ale kosztuje (osoba sprawdzająca potrzebuje pewnej przestrzeni na wykonywanie swoich obowiązków, trzeba ją chronić przed osobami trzecimi i trzeba jej zapłacić za wykonywaną pracę).

2.6. Metody nie wymagające wprowadzania danych przez użytkownika

Należałoby tutaj dla porządku przynajmniej wspomnieć, iż w systemach informatycznych stosuje się również rozpoznawanie tożsamości użytkownika np. na podstawie numeru identyfikacyjnego/seryjnego urządzenia, którym się posługuje. Gromadzenie danych pozwalających na identyfikację jest dokonywane bez wyraźnego działania użytkownika (czy tego chce czy nie). Jest to sposób mało wiarygodny, ale czasem przynoszący bardzo dobre rezultaty.

I tak np. w sieci Internet identyfikatorem może być adres IP, który musi być wykorzystywany z uwagi na sposób działania współczesnych sieci komputerowych. Jeżeli użytkownik dokonuje zakazanych prawnie czynności z komputera podłączonego do łącza stałego w jego domu, organy ścigania na podstawie danych uzyskanych od operatora telekomunikacyjnego mogą ustalić adres, a następnie wskazać konkretną osobę. Ale ten sam adres może być używany np. przez firmę, do sieci której ktoś się podłączył bezprzewodowo (i wtedy tak naprawdę nie ma możliwości jego odnalezienia).

Kolejnym przykładem mogą być żółte kropki drukowane przez różne kolorowe drukarki laserowe (pozwalać one na identyfikację np. numeru seryjnego i modelu urządzenia⁸⁵, a więc mogą istotnie pomóc w znalezieniu jego nabywcy i użytkowników) czy też cyfrowe telefony komórkowe (są one często kupowane w promocjach - ich numery identyfikacyjne są wtedy powiązane w dokumentacji operatorów z danymi konkretnych osób).

Innym rozwiązaniem jest pozostawianie swego rodzaju „znacznika” na komputerze użytkownika. Przykładem mogą być zawartości „ciasteczek” (cookies) pozostawiane w przeglądarkach www przez różne serwisy – tak długo, jak taki zbiór nie jest kasowany, witryna jest w stanie „stwierdzić”, że odwiedza ją ta sama osoba (jest to oczywiście słuszne założenie tylko wtedy, jeżeli z komputera korzysta jeden człowiek). W tym wypadku można się „bronić” wyłączając w przeglądarce obsługę „ciastek”. Dzieje się to jednak kosztem utraty pewnych funkcjonalności i wygody.

Można sądzić, że ta grupa będzie się rozszerzać. W wielu komputerach montuje się na stałe mikrofony i kamery cyfrowe. Nic nie stoi na przeszkodzie, aby oprogramowanie wykorzystywało je w różnych celach... Podobnie jest np. w telefonach komórkowych – z odpowiednim wewnętrznym oprogramowaniem mogą przecież służyć do podsłuchiwania użytkownika...

⁸⁵ <http://www.eff.org/Privacy/printers/>

3. Podsystem uwierzytelniania zdalnych użytkowników systemu informatycznego do wykorzystania w wydziale akademickim

Jak widać z poprzedniego rozdziału, istnieje dużo różnorodnych metod, które można zastosować w celu poznania i sprawdzenia tożsamości użytkowników. Z uwagi na rosnącą popularność Internetu i oczekiwania użytkowników przedstawione w podrozdziale 1.2 zdecydowano, że w ramach obecnego projektu zostanie przygotowane rozwiązanie umożliwiające przeprowadzenie uwierzytelniania w standardowej przeglądarce www (takiej jak Internet Explorer, Mozilla Firefox czy Opera). Wpierw zostanie przedstawiona koncepcja takiego rozwiązania, następnie szczegóły projektowe i informacje o gotowej implementacji.

3.1. Koncepcja

Zgodnie ze wstępem przedstawiona zostanie tutaj szczegółowo koncepcja rozwiązania w oderwaniu od konkretnej technologii i języka programowania. Będzie to zrobione odwrotnie niż w poprzednim rozdziale – najpierw określony zostanie sposób wprowadzania danych identyfikacyjnych, a dopiero później sam schemat uwierzytelniania.

3.1.1. Wybór sposobu wprowadzania danych identyfikacyjnych przez użytkownika

W podrozdziale 1.5 przyjęto, iż jednym z podstawowych kryteriów wyboru metody uwierzytelniania będzie jej ekonomiczność (rozumiana bardzo szeroko - przez pryzmat kosztów wdrożenia, ale również utrzymania i konserwacji). Z tego powodu odrzucone zostały wszystkie rozwiązania związane z cechami użytkowników oraz prawie wszystkie metody wymagające posiadania przez nich specjalnych przedmiotów (konkretnie: klucze sprzętowe, TPM, tokeny, karty stykowe i zbliżeniowe). Wymagają one bowiem instalacji specjalnych czytników w każdym miejscu (czyli zestawie komputerowym), z którego może skorzystać użytkownik albo przekazaniu każdemu z nich unikalnego przedmiotu (co stwarza dodatkowe problemy z ich dystrybucją).

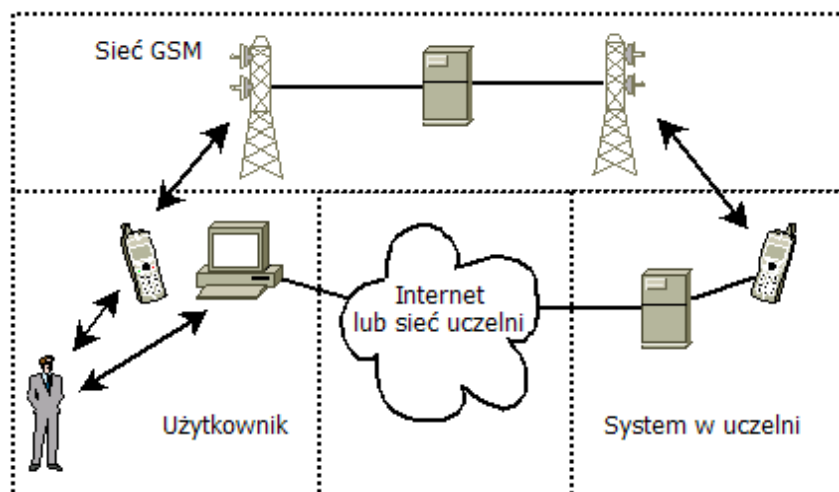
Ze względu na łatwość użytkowania odrzucone zostanie również wprowadzanie danych identyfikacyjnych z plików. Wymagałoby ono bowiem posiadania przez osobę uwierzytelnianą pamięci z nimi (dyskietki, płyty, itp.).

Zostaną wykorzystane hasła wielokrotne. Z uwagi na bezpieczeństwo będą one połączone z hasłami jednorazowymi. Osoba uwierzytelniana nie będzie jednak musiała nosić przy sobie listy ani tokena - wystarczy, że prześle do systemu niektóre z podanych jej danych ze swojego telefonu komórkowego. Nie powinno to być zbyt dużym ograniczeniem:

- urządzenie takie można teraz kupić niedrogo
- coraz więcej ludzi się z nim po prostu nie rozstaje (gdyż zaczyna spełniać coraz więcej ról – aparatu fotograficznego, odtwarzacza MP3, itd.)

- jego utrzymanie jest coraz tańsze

Pozostała część danych niezbędnych do uwierzytelnienia przekazywana będzie podobnie jak w innych systemach (tzn. siecią komputerową). Takie rozwiązanie zostało wprowadzone choćby ze względu na to, że treść SMS może być mniej lub bardziej świadomie i legalnie archiwizowana przez operatora GSM (i stąd nie powinna zawierać kompletu informacji wystarczającego do zalogowania się).



Rysunek 32. Rozwiązanie wykorzystywać będzie dwa praktycznie niezależne kanały do podawania danych identyfikacyjnych

3.1.2. Schemat uwierzytelniania

Każda osoba będzie musiała pamiętać swój login (unikalny) i hasło (6 cyfrowe). Dane te będą zapisane w bazie systemu częściowo w sposób zaszyfrowany:

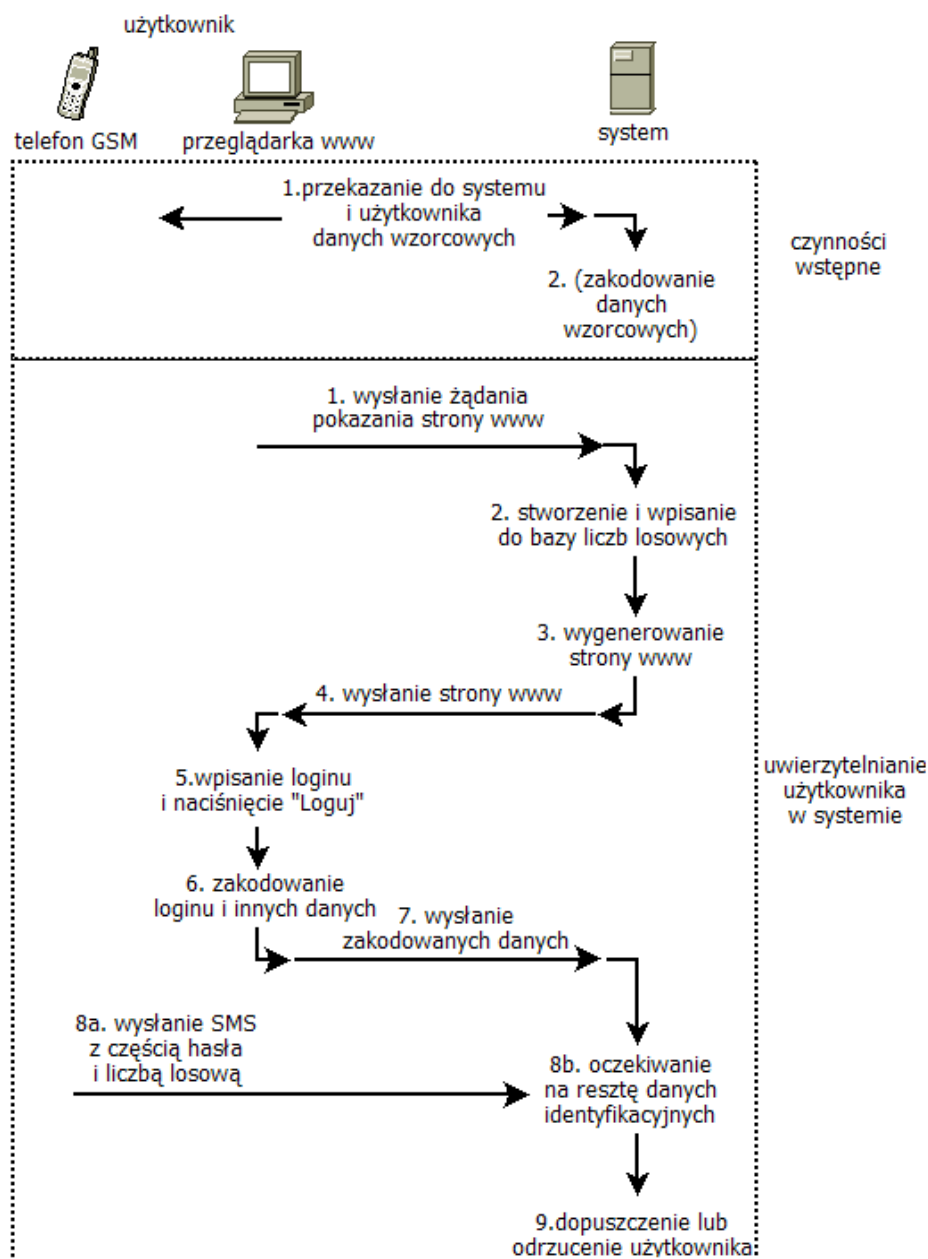
- login będzie zakodowany funkcją haszującą
- hasło będzie zapisane tekstem otwartym
- ciąg „numer telefonu+hasło” zostanie zakodowane funkcją haszującą

Użytkownik próbujący dokonać uwierzytelnienia będzie musiał wpisać w swojej przeglądarce www określony adres. Serwer otrzymujący żądanie wyświetlenia tej strony wraz z nią wygeneruje trzy liczby losowe (zostaną nazwane dalej A, B i C), jak również trzy liczby losowe z zakresu 1–6 (nazwane dalej D, E i F) oraz wartość logiczną prawda/fałsz (G). Dane te zostaną wpisane w bazie systemu wraz z datą upłynięcia ich ważności. Na stworzonej użytkownikowi stronie pojawią się:

- informacja o tym, jak wpisać treść SMS (będzie to dokładniej opisane dalej)
- klawiatura ekranowa służąca do wprowadzenia loginu (przy czym, żeby wyeliminować pomyłki, na ekranie komputera może być pokazywany jedynie ostatni wpisany znak)

W tym momencie użytkownik powinien wysłać z telefonu komórkowego SMSa o treści zawierającej liczbę losową B i po niej trzy znaki ze swojego hasła (określały je liczby losowe D, E i F). Jeżeli wartość logiczna G miała wartość fałsz, w pierwszej treści mają się pojawić znaki z hasła, później liczba B.

Kolejną czynnością wykonaną przez osobę uwierzytelnianą powinno być wpisanie loginu i naciśnięcie pokazanego obok klawiatury przycisku „Loguj”. Login zostanie następnie lokalnie (po stronie przeglądarki www użytkownika) zakodowany funkcją haszującą, do wyniku zostanie dołączona pierwsza liczba losowa (A), całość ponownie zostanie poddana działaniu funkcji haszującej i dopiero rezultat wraz z niezakodowaną liczbą C zostanie odesłany do serwera.



Rysunek 33. Schemat uwierzytelniania projektowanego podsystemu (jest to modyfikacja schematu „wyzwanie-odpowiedź”)

Po otrzymaniu całości (użytkownik ma na to określony czas) system będzie wykonywał kilkustopniowe sprawdzanie poprawności danych (żeby przejść do kolejnego punktu, wynik z poprzedniego musi być pozytywny):

- na podstawie C system poszuka ważnych A i B
- kolejno do wszystkich skrótów loginów zarejestrowanych użytkowników zostanie dodana liczba A, całość zostanie poddana działaniu funkcji haszującej i

porównana z rezultatem z przeglądarki www (musi być znaleziony przynajmniej jeden pasujący rekord)

- dla znalezionego w bazie zarejestrowanego użytkownika zostanie pobrane jego hasło oraz numer GSM i zostanie sprawdzone, czy z jego numeru wysłano SMS (musi być dokładnie jeden)
- sprawdzony zostanie numer centrum SMS wiadomości (czy jest jednym z numerów używanych przez polskich operatorów GSM)
- sprawdzone zostanie, czy w treści SMS otrzymano liczbę B i odpowiednie znaki z treści hasła przyporządkowanego do znalezionego w bazie zarejestrowanego użytkownika

Jeżeli wszystko skończy się sukcesem i znaleziona zostanie dokładnie jedna pasująca kombinacja, użytkownik zobaczy informację o pozytywnej weryfikacji, jak również datę i godzinę ostatniego nieudanego/udanego uwierzytelnienia.

3.1.3. Wady i zalety

Zaproponowane rozwiązanie cechuje się wieloma zaletami:

- login użytkownika przesyłany między przeglądarką i serwerem jest za każdym innym (dodatkowa zaleta: nie jest podawany otwartym tekstem). Żeby móc poznać jego skrót, osoba atakująca musi przechwycić transmisję w dwie strony: system -> użytkownik (w celu poznania A) i użytkownik -> system (w celu poznania skrótu z tekstu „hasz loginu + liczba A”). Nawet jeżeli będzie on znany osobie atakującej, jest wymagana jeszcze możliwość przesłania SMS z określonego numeru telefonu i treść hasła.
- część nowoczesnych telefonów komórkowych (choćby produkty Nokii) przechowuje w pamięci treść wysłanych sms – ponieważ B zmienia się za każdym razem, jak również nie wiadomo, które znaki z hasła zostały użyte konkretnym razem, nawet przeczytanie takiego SMS nie da nic potencjalnemu włamywaczowi (chyba, że będzie miał dostęp do treści SMS i strony www wyświetlonej użytkownikowi)
- login nie może być przechwycony przez keyloggery
- hasło nie jest przesyłane nigdy siecią komputerową
- w bazie systemu numer telefonu i login są zapisane w formie zaszyfrowanej
- zmiana hasła będzie wymuszała również podanie odpowiedniego numeru telefonu
- podjęta jest próba ograniczenia możliwości użycia jednego z serwisów internetowych do wysłania SMS z określonym numerem nadawcy

Propozycja ma również pewne ograniczenia:

- serwer odbierający SMS może mieć określoną wydajność i z tego powodu uwierzytelnianie zbyt wielu osób na raz może być niemożliwe
- hasło jest zapisywane w bazie systemu w sposób niezaszyfrowany (można zmniejszyć częściowo niebezpieczeństwo z tego wynikające odpowiednio tworząc implementację, co zostanie pokazane w kolejnym podrozdziale)
- konieczna jest pewna moc obliczeniowa po stronie przeglądarki www i serwera systemu

- użytkownik płaci za wysłanie SMS (choć z drugiej strony jest to pewna zaleta bardzo przydatna w systemie, w którym uwierzytelnianie służy do inicjacji przez użytkownika jakiejś pracochłonnej operacji wykonywanej przez pracowników instytucji – może bowiem zmniejszyć ilość tzw. „głupich żartów”)

Trzeba tutaj wspomnieć o problemie innego rodzaju. Operatorzy GSM w umowach podpisywanych z użytkownikami kart SIM umieszczają zastrzeżenia dotyczące sposobu ich wykorzystania. Poniżej przykład dla aktywacji zakupionych w promocji „Najtaniej do najdroższych bez telefonu” sieci Era:

„Niedozwolone jest stosowanie kart SIM aktywowanych w taryfie Era Nowy Komfort w urządzeniach realizujących funkcje zakończenia sieci stałej (Fixed Cellular Terminal, zwanych dalej „Urządzeniami FCT”) lub działających na podobnej zasadzie. W urządzeniach FCT lub działających na podobnej zasadzie Abonent może używać wyłącznie kart SIM aktywowanych na taryfach specjalnych przeznaczonych dla urządzeń typu FCT. Abonent nie ma również prawa za pomocą kart SIM, działających w sieci Era, kierować do sieci Era lub innych sieci telekomunikacyjnych ruchu z sieci innych operatorów, czerpiąc z tego tytułu bezpośrednio bądź pośrednio korzyści majątkowe lub przysparzając korzyści majątkowe osobom trzecim.”⁸⁶

Podobnie jest to zresztą formuowane w przypadku kart prepaid (taryfa Happy w ofercie Tak-Tak):

„Stosowanie urządzeń realizujących funkcje zakończenia sieci stałej (Fixed Cellular Terminal, zwanych dalej „Urządzeniami FCT”) lub działających na podobnej zasadzie, w których wykorzystywane będą karty SIM, działające w sieci Era, wymaga uzyskania od PTC Sp z o.o. pisemnej zgody. W urządzeniach FCT lub działających na podobnej zasadzie KLIENT może używać wyłącznie kart SIM aktywowanych na taryfach specjalnych przeznaczonych dla urządzeń typu FCT. Użytkownik nie ma prawa za pomocą kart SIM, działających w sieci Era, kierować do sieci Era lub innych sieci telekomunikacyjnych ruchu z sieci innych operatorów.”⁸⁷

oraz w sieciach innych niż Era. Można mieć oczywiście wątpliwości, czy w omawianym rozwiązaniu będziemy mieć do czynienia z tego typu sytuacją i autor nie jest w stanie podać żadnego przypadku, w którym operator powołał się na taki punkt, niemniej jednak możliwość taka istnieje. W tym momencie widać również przewagę karty typu prepaid – użytkownikowi oprócz zmiany numeru nie mogą grozić żadne inne sankcje (typu kary umowne).

3.2. Projekt

W tym podrozdziale umieszczone zostaną elementy projektu informatycznego podsystemu uwierzytelniania, którego koncepcja została przedstawiona wcześniej. Pełna ścieżka projektowa zgodna z jedną z metodyk (wraz ze wszystkimi przewidzianymi tam dokumentami) i szczegółowe opisy zostaną tutaj pominięte z uwagi m.in. na niewielki rozmiar projektu. Z uwagi na przejrzystość zrezygnowano z elementów pominiętych w koncepcji, a dodawanych w tego typu podsystemach

⁸⁶ http://www.era.pl/repositories/era_pl_repo1/documents/ind/ndn_bez.pdf w dniu 11 kwietnia

⁸⁷ http://www.era.pl/repositories/era_pl_repo1/documents/ind/ett_taryfa_Happy7.pdf w dniu 11 kwietnia

(takich jak np. wprowadzenie okresu ważności haseł, ocena ich „jakości” w trakcie ich ustalania albo blokada logowania się tego samego użytkownika z kilku miejsc).

3.2.1. Studium wykonalności

W rozdziale 1 wskazano na konieczność stosowania ograniczeń w dostępie do danych w instytucji edukacyjnej (w szczególności w wydziale akademickim). Następnie wskazano na uwierzytelnianie jako jeden ze środków pozwalających na uzyskanie tego celu oraz (w rozdziale 2) przedstawiono szeroko stosowane w systemach informatycznych jego rodzaje i metody. Ostatnim krokiem było opisanie koncepcji określonego podsystemu uwierzytelniania. Podsystem ten powinien być interesujący z co najmniej trzech ważnych względów:

- może pozwalać na uwierzytelnianie użytkowników w sposób bardziej wiarygodny niż najczęściej stosowane metody i być tańszy w eksploatacji niż rozwiązania wymagające instalowania dodatkowego sprzętu na każdym stanowisku komputerowym
- jego użycie nie powinno zbyt trudne dla przeciętnego użytkownika
- istnieją środki techniczne do jego realizacji

Z powyższych powodów zaleca się jego implementację i przynajmniej testowe sprawdzenie przydatności w warunkach rzeczywistych (wykorzystanie w jednej z aplikacji www instytucji).

Należy przypomnieć, że każda operacja uwierzytelniania wymaga wysłania SMS przez użytkownika (może to kosztować określoną kwotę) i stąd podsystem ten powinien być wykorzystany w aplikacji, w której czynność uwierzytelniania nie jest zbyt częsta (chodzi o to, aby użytkownicy byli w stanie zaakceptować wielkość związanych z tym płatności). Na działanie systemu mogą również wpłynąć zapisy umów, które są podpisywane z operatorem sieci GSM udostępniającym kartę SIM (dotyczy to zarówno abonamentów jak i usług prepaid). Niezbędne jest również utrzymywanie takiej karty (koszt ten musi ponieść instytucja). Możliwe jest istotne zwiększenie bezpieczeństwa rozwiązania przez dodatkowe używanie protokołu SSL (w szczególności wraz z płatnym certyfikatem kwalifikowanym).

Do implementacji niezbędna jest:

- wiedza programistyczna dotycząca tworzenia aplikacji dostępnych w przeglądarce www
- znajomość aplikacji, do której będzie dodawany projektowany podsystem
- wiedza związana z technologią SMS (dotycząca przynajmniej gotowych narzędzi)

3.2.2. Specyfikacja wymagań

Poniżej zostaną krótko przedstawione różne rodzaje wymagań stawianych projektowanemu podsystemowi.

3.2.2.1. Wymagania funkcjonalne

Projektowany podsystem ma zabezpieczać dostęp do aplikacji użytkowanych w instytucji edukacyjnej. Jego celem jest przeprowadzenie uwierzytelniania wszystkich osób, które chcą uzyskać dostęp do tej aplikacji. Uwierzytelnianie ma umożliwić jednoznaczne sprawdzenie, czy z aplikacji próbuje skorzystać jeden z zarejestrowanych użytkowników czy osoba trzecia niezwiązana z aplikacją.

Użytkownicy będą mieli styczność z podsystemem poprzez formularz do logowania do aplikacji. W ramach podsystemu konieczne jest również stworzenie oddzielnego elementu pozwalającego administratorowi systemu (aplikacji) na łatwe dodawanie zarejestrowanych użytkowników. Świadomie nie określono

1. wyglądu tych elementów
2. sposobu informowania użytkownika o błędach we wprowadzanych danych

pozostawiając pełną dowolność programistom.

Podsystem ma przechowywać dane o użytkownikach w bazie danych. Trzeba uwzględnić, że może się ona znajdować na innym stanowisku komputerowym niż sam podsystem uwierzytelniania. Połączenie między podsystemem i bazą danych oraz pomiędzy przeglądarką użytkownika i serwerem udostępniającym aplikację ma być nawiązane z użyciem protokołów TCP/IP.

3.2.2.2. Wymagania pozafunkcjonalne

Użytkownicy muszą mieć możliwość wykonania operacji uwierzytelniania ze stanowiska komputerowego, które ma dostęp do żądanej aplikacji. Konieczne jest zablokowanie możliwości wpisania loginu z klawiatury, natomiast niezbędne jest użycie myszy.

Nie przewidziano zgodności tworzonego podsystemu z żadnym dokumentem i wytycznymi przyjętymi w konkretnym istniejącym wydziale akademickim i nie określono technologii wykonania innych modułów podsystemu niż moduł uwierzytelniania (brak jest również założenia dotyczącego wykorzystanej bazy danych).

Względem modułu uwierzytelniania przyjęto następujące założenia:

1. ma być napisany w języku PHP
2. wskazane byłoby, aby w jego wnętrzu nie było fragmentów kodu źródłowego, które wymusiłyby publikację kodu źródłowego modułu uwierzytelniania lub fragmentów aplikacji www z nim powiązanych (wyklucza to np. wykorzystanie w nim elementów udostępnianych na licencji GNU GPL w wersji 2 z uwagi na jej punkt 2b⁸⁸)
3. musi istnieć możliwość przeprowadzenia uwierzytelniania z jego użyciem w przeglądarce Mozilla FireFox 2.x⁸⁹, Microsoft Internet Explorer 7.x⁹⁰ i Opera 9.x⁹¹ mających włączony język skryptowy JavaScript i działających w systemie operacyjnym Microsoft Windows XP z Service Pack 2

⁸⁸ dostępna w oryginale np. na stronie <http://www.gnu.org/licenses/gpl.html>

⁸⁹ <http://www.mozilla.com>

⁹⁰ <http://www.microsoft.com/ie>

⁹¹ <http://www.opera.com>

4. trzeba umieścić w jego kodzie elementy zmuszające przeglądarkę www do usuwania strony z własnej pamięci cache
5. nazwy zmiennych powinny odpowiadać nazwom podanym w koncepcji rozwiązania

Nie określono różnych innych wymagań takich jak:

- wielkościowe (np. ograniczenia na zużycie pamięci dyskowej przez dane identyfikacyjne związane z jednym użytkownikiem)
- niezawodnościowe (np. średni czas bezawaryjnego działania)
- wydajnościowe (np. liczba użytkowników, którzy mogą być uwierzytelniani w przeciągu minuty czy szybkość przeprowadzenia operacji uwierzytelniania)
- łatwość użytkowania (np. czas przeszkolenia administratora)

i inne

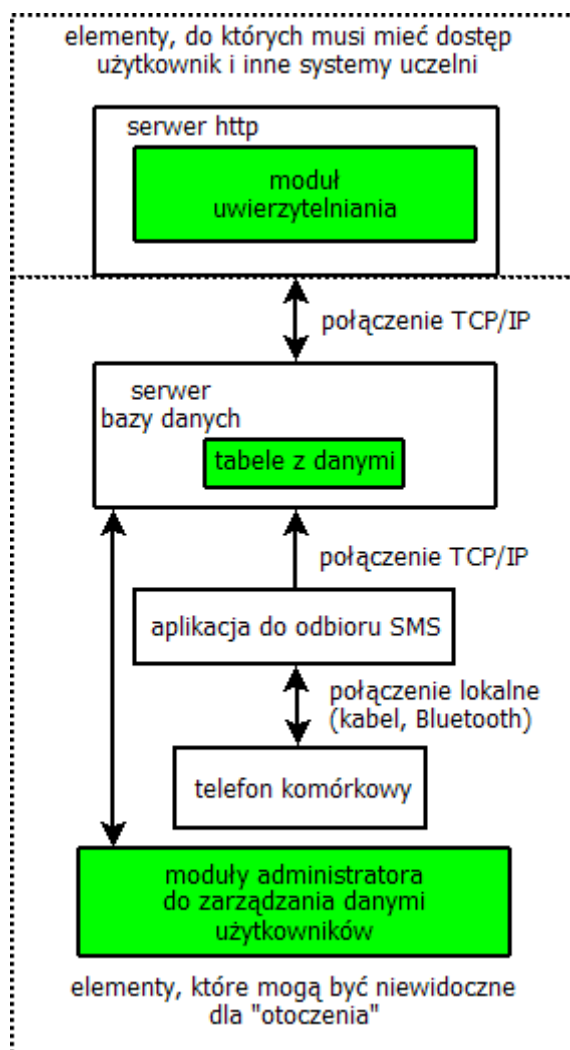
3.2.3. Architektura podsystemu

Na rysunku 34 przedstawiono architekturę proponowanego podsystemu uwierzytelniania. Zgodnie z nim serwer http, bazy danych i aplikacja do odbioru SMS mogą być umieszczone na oddzielnych komputerach (jest to więcej niż podane w wymaganiach). Warto jednak zauważyć i zaproponować, aby w praktycznej implementacji ze względów bezpieczeństwa użytkownicy dokonujący uwierzytelniania z wykorzystaniem projektowanego podsystemu mieli możliwość nawiązania połączenia TCP/IP wyłącznie z komputerem, na którym działa serwer http. Z podobnych względów (bezpieczeństwa) zaleca się również, aby aplikacja do odbioru SMS mogła jedynie dodawać nowe dane do bazy danych (bez możliwości ich kasowania i zmiany).

Zgodnie z przedstawionym zarysem w ramach projektu przygotowane zostaną trzy fragmenty:

1. moduł uwierzytelniania (który będzie mógł być zintegrowany z innymi aplikacjami www wykorzystywanymi w uczelni)
2. moduły administratora
3. struktura bazy danych

Jako pozostałe elementy (aplikacja do odbioru SMS, serwer www i serwer bazy danych) wykorzystane zostaną istniejące aplikacje. Kwestia używania SSL jest kwestią ich konfiguracji i stąd nie będzie dalej omawiana.



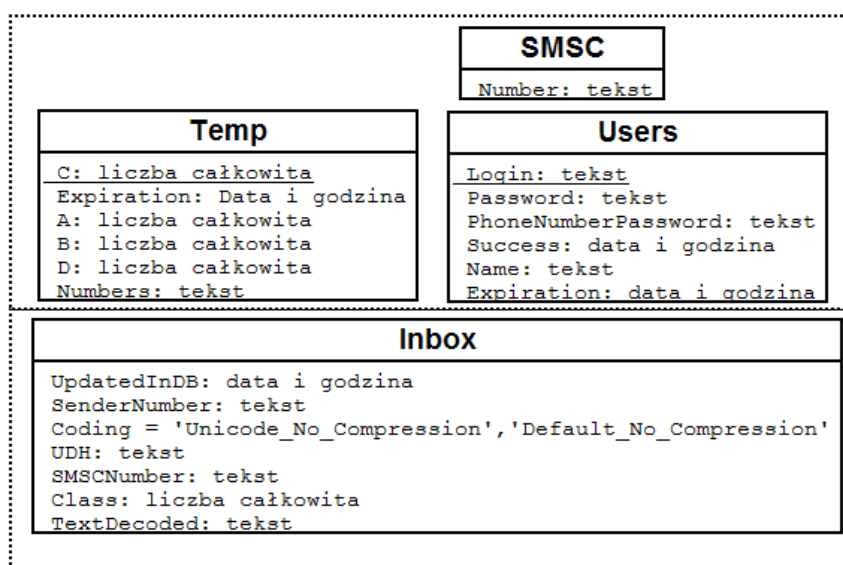
Rysunek 34. Zarys architektury projektowanego podsystemu uwierzytelniania (kolorem zielonym oznaczono stworzone samodzielnie elementy)

3.2.3.1. Struktura bazy danych

Poniżej przedstawiony został ogólny model bazy danych. Pomiedzy tabelami nie ma powiązań, stąd zrezygnowano z zamieszczania oddzielnego modelu konceptualnego i fizycznego. W projektowanym rozwiązaniu dane składowane będą w tabelach dwóch baz danych:

- jedna związana będzie bezpośrednio z odbiorem SMS
- druga przechowywać będzie pozostałe dane (informacje o użytkownikach, numerach SMSC i dane tymczasowe)

Nie jest to podyktowane kwestiami bezpieczeństwa (w wielu serwerach baz danych możliwe jest ustawienie praw dostępu nawet na poziomie poszczególnych pól), ale raczej związane z wygodą (jeżeli założymy, że w bazie z tabelą Inbox będziemy przechowywać również inne tabele wymagane przez aplikację do odbioru SMS, to rozwiązanie to może pozwalać na łatwiejszą aktualizację tej aplikacji).



Rysunek 35. Struktura bazy danych

Omówione zostaną teraz wykorzystywane tabele:

Tabela 2. Tabela SMSC przechowuje informacje o numerach centrum SMS, które mogą być wykorzystywane przez użytkowników do wysłania SMS

Nazwa pola	Rodzaj	Opis
Number	Tekst	Numer centrum SMS

Tabela 3. Tabela temp zawiera dane tymczasowe generowane w trakcie otwierania przez użytkowników stron logowania

Nazwa pola	Rodzaj	Opis
C	Liczba	Liczba losowa przedstawiona w koncepcji rozwiązania (zarazem klucz własny tabeli)
Expiration	Data i godzina	Czas, w którym wygenerowane dane stracą ważność
A, B, D	Liczby	Liczby losowe omówione w opisie koncepcji rozwiązania
Numbers	Tekst	Informacja o tym, które cyfry hasła mają być umieszczone w treści SMS

Tabela 4. Tabela users zawiera dane o użytkownikach systemu

Nazwa pola	Rodzaj	Opis
Login	Tekst	Login przetworzony funkcją haszującą (zarazem klucz własny tabeli)
Password	Tekst	Hasło w postaci jawnej
PhoneNumberPassword	Tekst	Numer telefonu wraz z hasłem przetworzone funkcją haszującą
Success	Data i godzina	Kiedy użytkownik był ostatnio zalogowany
Name	Tekst	Imię i nazwisko użytkownika
Expiration	Data i godzina	Do kiedy użytkownik może przebywać w systemie

Tabela 5. Tabela inbox zawiera treść odebranych SMS

Nazwa pola	Rodzaj	Opis
UpdatedInDB	Data i godzina	Czas, w którym wpisano SMS do bazy
SenderNumber	Tekst	Numer nadawcy wiadomości
Coding	Jedna z konkretnych wartości	Informacje o wykorzystanym w treści wiadomości kodowaniu. Rozwiązanie przyjmuje wiadomości w Unicode (wartość <i>Unicode_No_Compression</i>) oraz standardowym alfabecie wykorzystywanym w SMS (wartość <i>Default_No_Compression</i>) ⁹²
UDH	Tekst	Wartość nagłówka danych użytkownika (User Data Header). Rozwiązanie wymaga, aby był pusty. ⁹²
SMSCNumber	Tekst	Numer centrum SMS
Class	Liczba	Klasa wiadomości (rozwiązanie wymaga, aby SMS nie miał jej ustawionej, co będzie tutaj odpowiadać wartości <i>-1</i>) ⁹²
TextDecoded	Tekst	Zdekodowana treść wiadomości SMS

3.2.3.2. Moduł uwierzytelniania

Sposób wytworzenia tego elementu ściśle powiązany z technologią, w jakiej wykonano aplikację (w której dokonujemy uwierzytelnienia). Zgodnie z wymaganiami pozafunkcjonalnymi moduł ten ma być wykonany w języku skryptowym PHP.

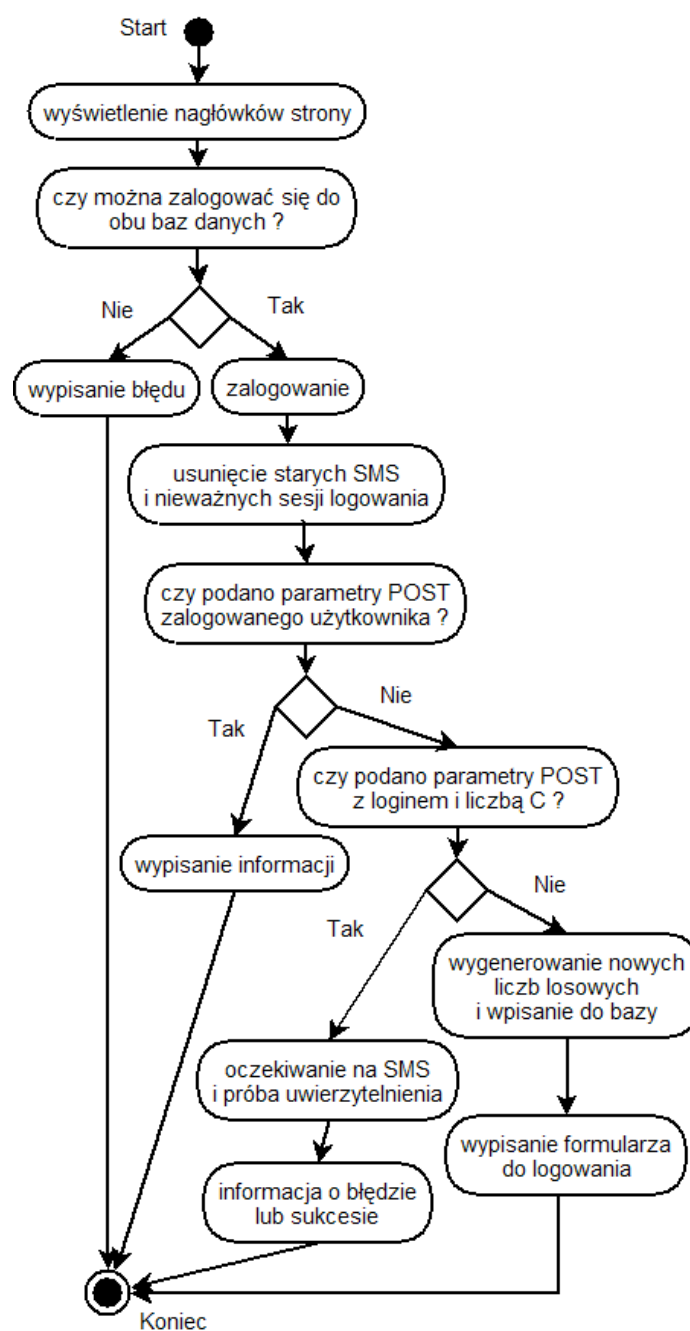
Przyjęto, że całość zostanie zawarta w jednym pliku (będzie to jedna strona www). Jej schemat blokowy (elementy kodu PHP) został przedstawiony na rysunku 36. Ponieważ jest on ogólny, nie zawiera co najmniej trzech ważnych elementów i informacji (które muszą być uwzględnione):

- w formularzu do logowania konieczne jest zakodowanie wpisanych danych po podaniu ich przez użytkownika i przed ich wysłaniem do serwera http. Zgodnie z wymaganiami można wykorzystać w przeglądarce www język JavaScript. Pozwoli on na obsługę zdarzenia *onSubmit*⁹³ formularza i wywołanie w nim procedury kodującej dane.
- wymagania określają, że nie można wpisywać loginu z klawiatury ani przysyłać jego kolejnych znaków otwartym tekstem do serwera http. Powoduje to, że konieczne jest znów użycie np. JavaScript – odpowiednia procedura w tym języku będzie reagować na zdarzenie *OnClick* „klawiszy” klawiatury ekranowej dopisując przypisane do nich „znaki” do pola przechowującego login w formularzu

⁹² Szerzej kwestie budowy SMS i celowość umieszczania tutaj konkretnych parametrów zostały przedstawione np. w kodzie pakietu Gammu dostępnego na stronie <http://www.gammu.org>, podstawą do jego napisania były zaś m.in. specyfikacje GSM dostępne na stronie <http://pda.etsi.org/pda/queryform.asp> (głównie kolejne wydania dokumentów oznaczanych jako GSM 03.38 i GSM 03.40)

⁹³ Model obiektowy implementowany we współczesnych przeglądarkach www jest szczegółowo opisywany w wielu poradnikach w Internecie i nie ma większego sensu go tutaj omawiać

- ze względów bezpieczeństwa (jak najmniejsze możliwości nadużyć związanych z wygenerowanymi liczbami losowymi) w punkcie „wyświetlenie nagłówków strony” konieczne jest dodanie funkcji wymuszającej przeładowywanie strony co jakiś czas w przeglądarkach z JavaScript oraz odpowiedniego nagłówka, który mógłby być użyty, gdy przeglądarka użytkownika nie ma tego języka (wtedy formularz logowania nie powinien się pojawiać w ogóle). Dodatkowo trzeba wykorzystać nagłówki zgodne ze specyfikacją HTTP⁹⁴, aby zawartość strony nie była przechowywana w pamięci cache przeglądarki www. Z tych samych względów wybrano metodę POST (a nie GET) do wysłania danych z formularza logowania.



Rysunek 36. Schemat blokowy modułu uwierzytelniania

W celu uproszczenia rozwiązania zaproponowano:

⁹⁴ „RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1”, The Internet Society, czerwiec 1999

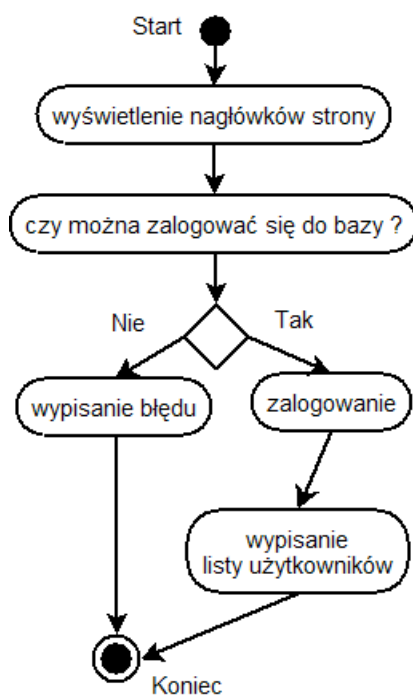
- zrezygnowanie z przedstawiania dokładniejszej struktury kodu (m.in. tworzenia diagramów klas, komponentów, obiektów czy też pakietów)
- przedstawienie sposobu „współdziałania” kodu z użytkownikiem w formie schematu uwierzytelniania (zaprezentowany w podrozdziale 3.1.2 pełni rolę diagramu sekwencji) oraz przypadków testowych (podrozdział 3.3.3), brak np. diagramów przypadków użycia czy kooperacji

Należy wyjaśnić również, iż w module tym dodatkowo (poza elementami wymaganymi w projekcie) umieszczono fragmenty związane z obecnością zalogowanego użytkownika w systemie takie jak wyświetlanie informacji i uaktualnianie pola Expiration w tabeli users (ta ostatnia czynność wykonywana jest po automatycznym przeładowywaniu strony co kilka sekund). Zostały one dodane na potrzeby prezentacji pracy dyplomowej i w rzeczywistym systemie prawdopodobnie powinny zostać inaczej zaimplementowane.

3.2.3.3. Moduły administratora

Z uwagi na obecność gotowej strony www z modulem uwierzytelniania napisanej w języku PHP zdecydowano się wykorzystać jej kod do napisania dwóch modułów administratora:

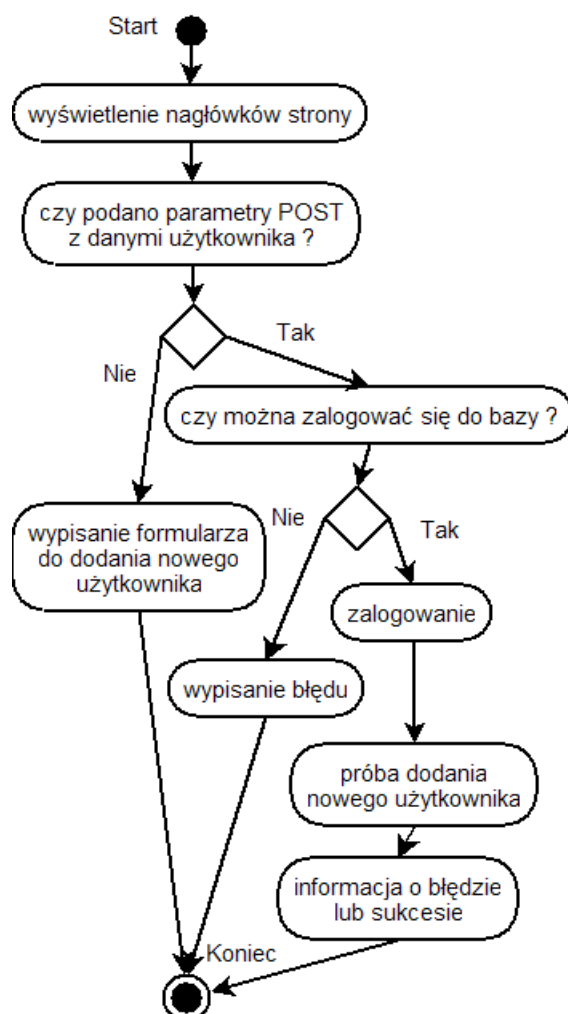
1. do dodawania nowych użytkowników do systemu (wyszczególniony w wymaganiach)
2. do pokazywania listy użytkowników w systemie wraz z informacją, czy są zalogowani w systemie (jako dodatkowy element)



Rysunek 37. Schemat modułu do wyświetlania listy użytkowników

Schemat modułu do dodawania nowych użytkowników jest podobny jak schemat strony z modulem uwierzytelniania (który został przedstawiony na

rysunku 36), podobnie jak tam zdecydowano się w ogóle nie wykonywać pewnych elementów projektowych.



Rysunek 38. Schemat modułu do dodawania nowych użytkowników

3.3. Implementacja

Poniżej przedstawiono informacje o powstałej w ramach pracy dyplomowej implementacji. Oprócz nich dużo szczegółów o wykorzystanych rozwiązaniach zawarto bezpośrednio w kodach źródłowych.

3.3.1. Wybór technologii

Do realizacji wykorzystano zestaw narzędzi typu Open Source:

- aplikację konsolową z pakietu Gammu⁹⁵ do odbierania SMS (świadomie jest to starsza wersja 1.09.16 jako ostatnia w pełni przygotowana przez autora)

⁹⁵ <http://www.gammu.org>

- serwer http Apache⁹⁶ (wersja 2.2.4) z językiem skryptowym PHP⁹⁷ (wersja 5.2.2) do wyświetlania użytkownikowi stron WWW
- serwer baz danych MySQL⁹⁸ do składowania danych (wersja 5.0.37)

Użyto również:

- kodu napisanego w JavaScript dotyczącego MD5 (którego autorem jest Henri Torgemane) i udostępnianego na własnej licencji. Wybór tej funkcji haszującej może wydać się dziwny (mając na względzie opisywane w rozdziale 2 informacje o znalezieniu w niej kolizji), ale został on podyktowany wyłącznie chęcią poszanowania praw autorskich (inne funkcje mogą być bowiem objęte takimi licencjami, że ich bezpłatne użycie jest niemożliwe)
- funkcji pozwalających zmniejszyć ryzyko ataków typu SQL Injection (w skrócie: wykorzystujących błędy w oprogramowaniu pozwalające na wykonywanie dowolnych komend SQL na bazach danych powiązanych z tym oprogramowaniem⁹⁹) opublikowane w serwisie hacking.pl przez Łukasza Lacha¹⁰⁰

3.3.2. Realizacja podsystemu

Zdecydowano, że tworzony moduł uwierzytelniania nie będzie zintegrowany z żadną aplikacją (ale pozostaje autonomicznym tworem).

Starano się spełnić wszystkie założenia projektowe. Można mieć jedynie zastrzeżenie, czy wypełniono zalecenie dotyczące możliwości nieudostępniania kodu źródłowego stworzonych modułów lub elementów z nimi współpracujących (przedstawionego w wymaganiach niefunkcjonalnych). Wynika to z zapisów licencyjnych bazy danych MySQL. Jeżeli będzie to istotny problem, zaleca się użycie np. bazy danych PostgreSQL¹⁰¹ (wymagana będzie zamiana w kodzie wszystkich wywołań dotyczących bazy oraz nowsza wersja Gammu).

Kolejna uwaga dotyczy modułu uwierzytelniania – w przypadku zajścia kolizji (wywołanie funkcji haszującej na różnych danych da w wyniku ten sam skrót i umożliwi ewentualnie uwierzytelnienie nieuprawnionego użytkownika) wyświetla informację o błędzie i wymusza ponowną próbę uwierzytelniania (z innymi danymi).

Warto zauważyć, iż zarówno w wymaganiach jak i w koncepcji nie ma ani słowa o sytuacji, gdy co najmniej dwóch użytkowników używa tego samego numeru telefonu komórkowego (implementacja dopuszcza taką możliwość).

Moduł uwierzytelniania zawarty został w pliku *loguj.php*, moduły administratora w plikach *dodaj.php* i *lista.php*. Korzystają one z funkcji w pliku *wspolne.php* oraz *md5.js*. Struktura bazy została wygenerowana z załączonego pliku *mgr2.sql*. Kody źródłowe tych plików zostały przedstawione w Załączniku A, tabela inbox bazy danych powstała natomiast na podstawie zbioru *mysql.sql* z pakietu Gammu.

⁹⁶ <http://www.apache.org>

⁹⁷ <http://www.php.net>

⁹⁸ <http://www.mysql.org>

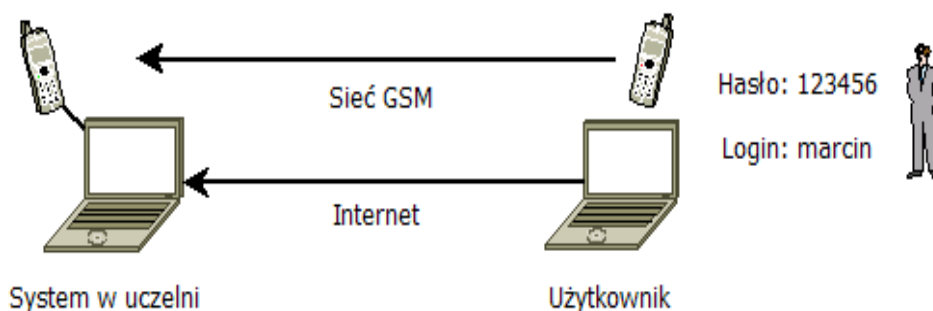
⁹⁹ pewne przykłady można obejrzeć np. na stronie WikiPedi http://pl.wikipedia.org/wiki/SQL_injection

¹⁰⁰ <http://hacking.pl/5845>

¹⁰¹ <http://www.postgresql.org>

3.3.3. Testowanie podsystemu

Środowisko testowe zostało przygotowane zgodnie ze schematem z rysunku 39 – jeden komputer pełnił rolę serwera uczelnianego (pracowała na nim baza MySQL, aplikacja do odbioru SMS i serwer http w wersjach przedstawionych wcześniej, był do niego również podłączony telefon komórkowy), kablem (symulującym transmisję TCP/IP przez Internet) połączono z nim drugi komputer (użytkownika), z którego przeprowadzano m.in. próby uwierzytelniania z podanym na rysunku loginem i hasłem.



Rysunek 39. Schemat środowiska testowego podsystemu uwierzytelniania

Oprogramowanie na serwerze zostało uruchomione w systemie Microsoft Windows XP, telefonem odbierającym SMS była Nokia 6230 (model RH-12 z wersją firmware 5.50 połączony kablem DKU-2). Aplikacje zostały pobrane bezpośrednio ze stron producentów i zainstalowane w następującej kolejności:

1. *apache_2_2_4-win32-x86-no_ssl.msi* (uruchomiony jako usługa)
2. *php-5.2.2-win32-installer.msi* (wybrano opcję ustawienia automatycznej konfiguracji serwera Apache 2.2 oraz instalację rozszerzenia MySQL)
3. *mysql-5.0.3.7-win32.zip* (oprócz instalacji jako usługa skopiowano do katalogu z Windows plik *libmysql.dll*)
4. *gammu_win32.zip* (instalacja polegała na rozpakowaniu do dowolnie wybranego katalogu)

Następnie:

- w pliku *php.ini* włączono rozszerzenie *php_mysql.dll*.
- pobrano i rozpakowano zawartość pliku *phpMyAdmin-2.10.1-english.tar.gz*¹⁰² i następnie w *config.default.php* ustawiono hasło użytkownika do bazy (należy zastrzec, że phpMyAdmin¹⁰² nie ma nic wspólnego z samym projektem i został jedynie wykorzystany w celu łatwiejszego zarządzania zawartością baz danych)
- utworzono dwie bazy danych programem phpMyAdmin, a następnie zaimportowano do pierwszej zawartość pliku *mgr2.sql* i do drugiej zawartość *mysql.sql*
- do odpowiedniego katalogu z Apache skopiowano pliki projektu (zarówno moduł uwierzytelniania jak i administratora)
- wpisano niezbędne ustawienia w pliku *wspolne.php* (takie jak numer telefonu czy hasło użytkownika wymagane przy dostępie do bazy danych)

¹⁰² <http://www.phpmyadmin.net/>

- ustawiono parametry w plikach konfiguracyjnych pakietu Gammu *gammurc* i *smsdrc* (takie jak rodzaj połączenia i hasło dostępu do bazy)
- uruchomiono Gammu w trybie odbioru SMS komendą ***gammu -smsd MYSQL smsdrc***

Na początku testów w przeglądarce www uruchomionej w komputerze użytkownika otworzono stronę służącą do dodawania użytkownika (pokazaną na rysunku 40) i dodano wspomnianego wcześniej użytkownika. Następnie w tej samej przeglądarce otworzono dwie strony:

1. z listą zalogowanych w systemie użytkowników (podobną jak na rysunku 41)
2. służącą do uwierzytelniania użytkownika (analogiczną do pokazanej na rysunku 42)

i dokonano operacji uwierzytelniania (w jej wyniku na stronie z listą zalogowanych użytkowników zaobserwowano zmianę informacji oraz otrzymano komunikat podobny do widocznego na rysunku 43).

Dodanie nowego użytkownika - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Dodanie nowego użytkownika

WIRTUALNY DZIEKANAT
Dodanie nowego użytkownika

Proszę wpisać login korzystając z klawiatury ekranowej oraz resztę poniższych danych z klawiatury komputera i nacisnąć przycisk Dodaj

Login

1 2 3 4 5 6 7 8 9 0 - = Del
q w e r t y u i o p []
a s d f g h j k l ; '
z x c v b n m . , / \

Imię i nazwisko

Hasło

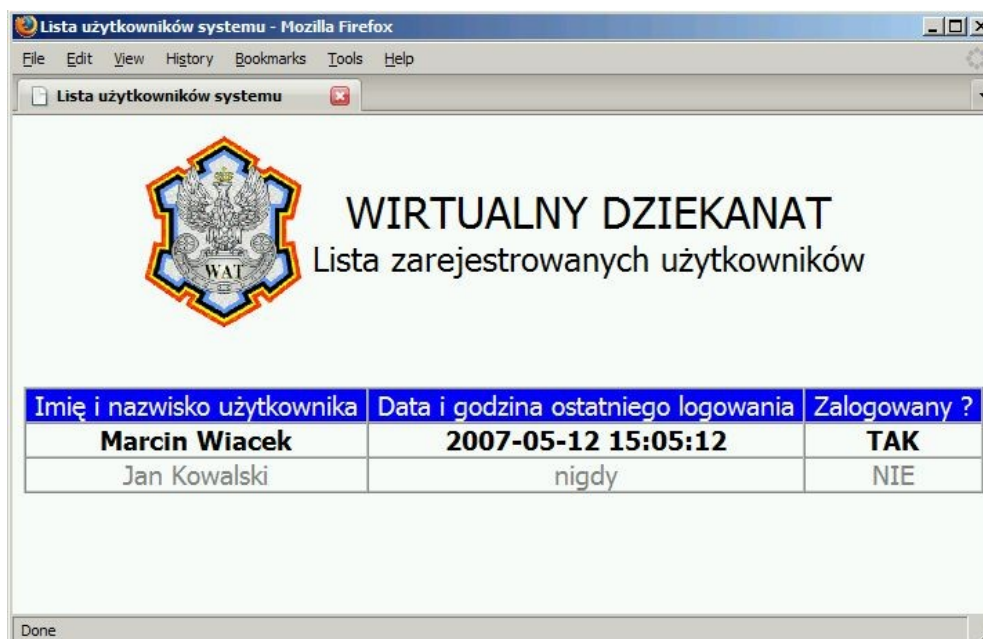
Powtórzenie hasła

Numer telefonu GSM
(format +48123456789)

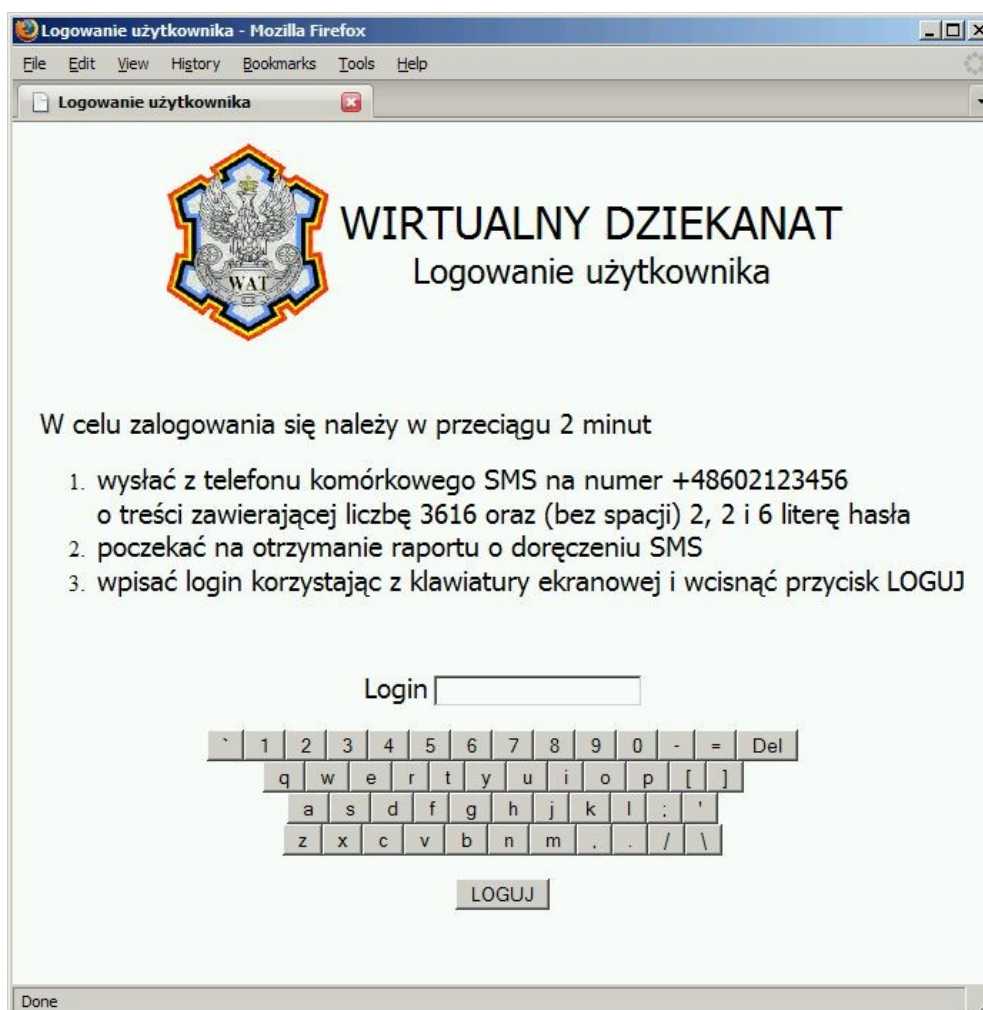
DODAJ

Done

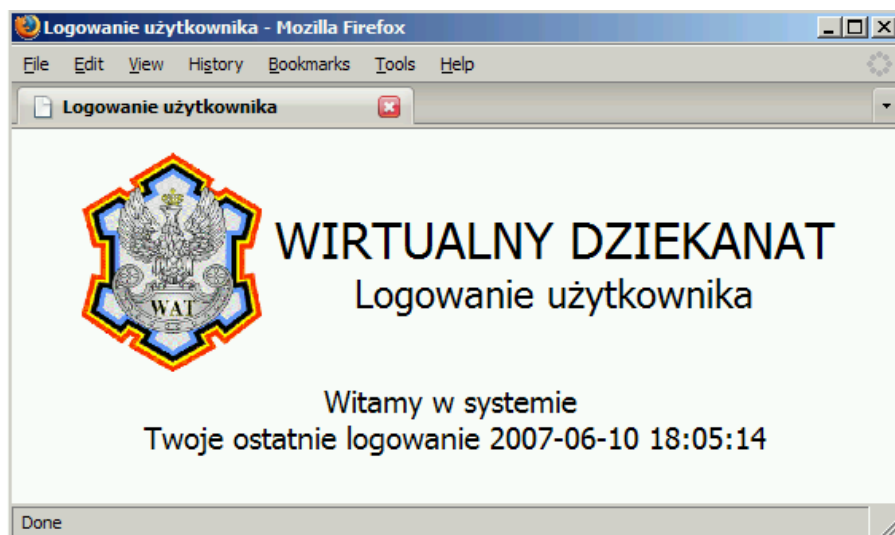
Rysunek 40. Wygląd zaimplementowanej strony do dodawania nowego użytkownika (przed wysłaniem danych)



Rysunek 41. Wygląd modułu do wyświetlania listy użytkowników



Rysunek 42. Wygląd zaimplementowanej strony www z modułem uwierzytelniania (przed zalogowaniem się użytkownika)



Rysunek 43. Informacja pokazywana użytkownikowi po poprawnym uwierzytelnieniu

Wykonano również w trzech wymaganych przeglądarkach bardziej szczegółowe testy akceptacyjne (zgodności z założeniami) modułu do dodawania nowych użytkowników i modułu uwierzytelniania. System w stanie początkowym (przed wykonaniem każdego testu) miał zarejestrowanego jednego użytkownika (o danych pokazanych na rysunku 39). Po każdym teście sprawdzano zawartość baz danych programem phpMyAdmin¹⁰² i listę użytkowników w module do ich wyświetlania oraz doprowadzano cały system do stanu początkowego.

Tabela 6. Przeprowadzone testy modułu administratora do dodawania nowych użytkowników

Przebieg testu	Rezultat
Otworzenie w przeglądarce www bez języka JavaScript	Brak możliwości dodania użytkownika
Otworzenie w przeglądarce www z językiem JavaScript	Wyświetlenie formularza umożliwiającego dodanie użytkownika
Próba wysłania do systemu formularza z przeglądarki z JavaScript bez wszystkich wymaganych danych (przez pozostawianie przynajmniej jednego z pól pustego i naciśnięcie przycisku Dodaj)	Wyświetlenie komunikatu o błędzie
Próba wysłania do systemu formularza z przeglądarki z JavaScript ze wszystkimi danymi, gdzie hasło i powtórzenie hasła są różne (przez wypełnienie pól formularza i naciśnięcie przycisku Dodaj)	Wyświetlenie komunikatu o błędzie
Próba wysłania do systemu formularza z przeglądarki z JavaScript ze wszystkimi danymi i zbyt krótkim hasłem (przez odpowiednie wypełnienie pól formularza i naciśnięcie przycisku Dodaj)	Wyświetlenie komunikatu o błędzie

Przebieg testu	Rezultat
<ol style="list-style-type: none"> 1. Próba wysłania do systemu formularza z przeglądarki z JavaScript ze wszystkimi danymi spełniającymi założenia (przez odpowiednie wypełnienie pól formularza i naciśnięcie przycisku Dodaj) 2. Próba wysłania do systemu formularza z przeglądarki z JavaScript ze wszystkimi danymi spełniającymi założenia, gdzie login jest ten sam, co w punkcie 1 testu (przez odpowiednie wypełnienie pól formularza i naciśnięcie przycisku Dodaj) 	Wyświetlenie informacji o sukcesie po pierwszym kroku testu i otrzymanie komunikatu o błędzie po drugim kroku

Tabela 7. Przeprowadzone testy modułu uwierzytelniania

Przebieg testu	Rezultat
<ol style="list-style-type: none"> 1. Otworzenie w przeglądarce www bez języka JavaScript 2. Odczekanie około 2 minut 	Brak możliwości uwierzytelnienia użytkownika (po punkcie 1) i przeładowanie strony (po punkcie 2)
<ol style="list-style-type: none"> 1. Otworzenie w przeglądarce www z językiem JavaScript 2. Odczekanie około 2 minut 	Wyświetlenie formularza umożliwiającego uwierzytelnienie użytkownika (po punkcie 1) i przeładowanie strony (po punkcie 2)
<ol style="list-style-type: none"> 1. Otworzenie w przeglądarce www z językiem JavaScript 2. Próba wpisywania loginu w polu przeglądarki z klawiatury 	Brak możliwości wpisania loginu z klawiatury
Próba wysłania do systemu formularza z przeglądarki z JavaScript bez loginu (przez pozostawienie pustego pola login i naciśnięcie przycisku Loguj)	Wyświetlenie komunikatu o błędzie
Próba wysłania do systemu formularza z przeglądarki z JavaScript z niezarejestrowanym loginem (przez wypełnienie pola login i naciśnięcie przycisku Loguj)	Wyświetlenie komunikatu o błędzie po około 20 sekundach
Próba wysłania do systemu formularza z przeglądarki z JavaScript z zarejestrowanym loginem (przez wypełnienie pola login i naciśnięcie przycisku Loguj)	Wyświetlenie komunikatu o błędzie po około 20 sekundach
<ol style="list-style-type: none"> 1. Próba wysłania do systemu formularza z przeglądarki z JavaScript z zarejestrowanym loginem (przez wypełnienie pola login i naciśnięcie przycisku Loguj) 2. Wysłanie SMS do systemu z poprawną treścią z niezarejestrowanego numeru telefonu 	Wyświetlenie komunikatu o błędzie po około 20 sekundach

Przebieg testu	Rezultat
<ol style="list-style-type: none"> 1. Próba wysłania do systemu formularza z przeglądarki z JavaScript z zarejestrowanym loginem (przez wypełnienie pola login i naciśnięcie przycisku Loguj) 2. Wysłanie SMS do systemu ze złą treścią z zarejestrowanego numeru telefonu 	Wyświetlenie komunikatu o błędzie po około 20 sekundach
<ol style="list-style-type: none"> 1. Próba wysłania do systemu formularza z przeglądarki z JavaScript z zarejestrowanym loginem (przez wypełnienie pola login i naciśnięcie przycisku Loguj) 2. Wysłanie SMS do systemu z poprawną treścią z zarejestrowanego numeru telefonu 	Wyświetlenie komunikatu o sukcesie i daty i godziny ostatniego logowania
<ol style="list-style-type: none"> 1. Próba wysłania do systemu formularza z przeglądarki z JavaScript z zarejestrowanym loginem (przez wypełnienie pola login i naciśnięcie przycisku Loguj) 2. Wysłanie SMS do systemu z poprawną treścią z zarejestrowanego numeru telefonu 3. Próba wysłania do systemu formularza z drugiego okna przeglądarki z JavaScript z tym samym loginem, co w punkcie 1 (przez wypełnienie pola login i naciśnięcie przycisku Loguj) 4. Wysłanie SMS do systemu z treścią jak w punkcie 2 z zarejestrowanego numeru telefonu 	Wyświetlenie komunikatu o sukcesie i daty i godziny ostatniego logowania (w punkcie 2) i komunikatu o błędzie (w punkcie 4 po około 20 sekundach)
<ol style="list-style-type: none"> 1. Usunięcie z tabeli smsc numeru używanego centrum SMS 2. Próba wysłania do systemu formularza z przeglądarki z JavaScript z zarejestrowanym loginem (przez wypełnienie pola login i naciśnięcie przycisku Loguj) 3. Wysłanie SMS do systemu z poprawną treścią z zarejestrowanego numeru telefonu 	Wyświetlenie komunikatu o błędzie po około 20 sekundach

Nie zaplanowano i nie wykonano natomiast m.in. testów wydajnościowych (np. wiele prób uwierzytelniania na raz w celu stwierdzenia jaka jest maksymalna liczba użytkowników, którzy mogliby być uwierzytelniani).

Podsumowanie

W ramach pracy wykonano wszystkie postawione zadania. Rozpoczęto od analizy potrzeb instytucji edukacyjnych w zakresie stosowania uwierzytelniania w systemach informatycznych. Przedstawiono typowe sytuacje i systemy, gdzie powinno być ono stosowane. Sprawdzono, jak wybrane kwestie przetwarzania danych i zabezpieczania ich przed nieautoryzowanym dostępem regulują różne polskie ustawy. Było to podstawą do stwierdzenia, iż tematyka ta jest rzeczywiście bardzo interesująca i ważna, a podsystem uwierzytelniania powinien być integrowalny z aplikacjami i usługami udostępnianymi poprzez przeglądarki www.

Kolejnym wykonanym krokiem było określenie kryteriów, jakie powinna spełniać idealna metoda uwierzytelniania i dokonanie z ich uwzględnieniem szczegółowego przeglądu dostępnych na rynku rozwiązań. Opisano tam wpierw możliwe do wykorzystania w uwierzytelnianiu narzędzia (takie jak funkcje skrótu, szyfrowanie symetryczne i asymetryczne), następnie wyróżniono cztery schematy przeprowadzania tego procesu (porównywania z danymi wzorcowymi, przekazywania „odpowiedzi” na „wyzwanie”, podpisu elektronicznego oraz SSL) i pokazano różne metody wprowadzania danych identyfikacyjnych do systemów (to właśnie ich różnorodność spowodowała, iż tematyka ta jest obecnie tak obszerna). W trakcie wykonywania opisanych czynności wyraźnie zauważono trzy prawidłowości:

- w wielu rozwiązaniach wysoka ich skuteczność może być uzyskana jedynie wtedy, gdy użytkownicy będą się stosować do pewnych określonych zasad. Ponieważ jednak wykorzystywane systemy nie wymuszają ich respektowania, uzyskane bezpieczeństwo często jest jedynie iluzoryczne.
- szczególnie wysoką skutecznością charakteryzują się rozwiązania sprzętowe (gdzie użytkownicy mogą nawet znać dokładnie metodę zabezpieczenia, ale po prostu nie mają możliwości technologicznych na jej złamanie). Problemem w ich upowszechnianiu są głównie kwestie finansowe oraz opór użytkowników.
- w różnych implementacjach uwierzytelnianie nie jest wprowadzone w sposób kompleksowy.

Mając m.in. to na względzie zaproponowano nową metodę uwierzytelniania, w której użytkownik wysyła do systemu część danych identyfikacyjnych tradycyjnie (poprzez sieć komputerową), a pozostałe z użyciem swojego telefonu komórkowego. Rozwiązanie to nie tylko łączy wykorzystanie haseł i loginów z noszonym praktycznie zawsze przez użytkownika przedmiotem, ale dodatkowo jest kompleksowe i wymusza stosunkowo wysoki poziom bezpieczeństwa. Nie powinno również wywoływać problemów u osób go wykorzystujących (jedyna innowacyjna czynność, czyli wysyłanie wiadomości SMS z telefonu, nie będzie raczej czymś trudnym dla jego właściciela) i co najważniejsze – być może pojawi się w różnych miejscach i instytucjach edukacyjnych, gdyż nie kosztuje zbyt dużo (choćaby dlatego, iż w przeciwieństwie do systemów znanych np. z banków internetowych to osoby uwierzytelniane są obciążane opłatami za wiadomości SMS).

Ostatnimi elementami pracy był projekt podsystemu uwierzytelniania wykorzystującego opisaną metodę oraz jego implementacja w języku PHP (której kod bezpośrednio można dołączyć do różnych aplikacji). Została ona gruntownie przetestowana, co pozwoliło potwierdzić słuszność dokonanych w projekcie założeń.

Bibliografia

Artykuły i prezentacje

1. Martin Lades, Jan C. Vorbrüggen, Joachim Buhmann, Jörg Lange, Christoph v.d. Malsburg, Rolf P. Würtz i Wolfgang Konen, „*Distortion Invariant Object Recognition in the Dynamic Link Architecture*”, IEEE transactions on computers, vol. 42, no. 3, marzec 1993
(<http://www.vision.caltech.edu/CNS179/papers/Lades93.pdf>)
2. Henry A. Rowley, Shumeet Baluja i Takeo Kanade, „*Neural NetworkBased Face Detection*”, School of Computer Science, Carnegie Mellon University, Pittsburgh, 1996 (<http://imagelab.ing.unimo.it/ttei/varie/rowley96neural.pdf>)
3. Carlos Morimoto, Dave Koons, Arnon Amir i Myron Flickner, „*Real-Time Detection of Eyes and Faces*”, IBM Research Center, USA, 1998
(<http://www.acm.org/icmi/1998/Papers/Morimoto.pdf>)
4. Rein-Lien Hsu, Mohamed Abdel-Mottaleb i Anil K. Jain, „*Face detection in color images*” (http://www.cg.cs.uni-bonn.de/docs/teaching/2002/WS/cv_hand_tracking/documents/papers/face-detection-in-color.pdf)
5. Simson L. Garfinkel i Abhi Shelat „*Remembrance of Data Passed: A Study of Disk Sanitization Practices*”, MIT, styczeń 2003
(http://www.computer.org/portal/cms_docs_security/security/v1n1/garfinkel.pdf)
6. Marcin Szeliga, „*NTLM cz.1*” i „*NTLM cz.2*”, Microsoft, maj 2003
(<http://www.microsoft.com/poland/technet/article/art001.msp> i <http://www.microsoft.com/poland/technet/article/art003.msp>)
7. Philippe Oechslin, „*Making a Faster Cryptanalytic Time-Memory Trade-Off*”, Laboratoire de Sécurité de Cryptographie (LASEC) Ecole Polytechnique Fédérale de Lausanne, Szwajcaria, maj 2003
(<http://lasecwww.epfl.ch/~oeechslin/publications/crypto03.pdf>)
8. Leszek Stępień, „*Dwanaście cech linii*”, Laboratorium Komendy Stołecznej Policji, Warszawa, 1 stycznia 2004
(http://www.ksp.waw.pl/laboratorium/download/Publikacje/Dwanascie_cek_li_nii.pdf)
9. Jerzy Karbowski, „*Ochrona i bezpieczeństwo danych w systemach informatycznych Podstawy prawne ochrony informacji w Polsce*”, WAT, Warszawa 2005
10. Jerzy Karbowski, „*Ochrona i bezpieczeństwo danych w systemach informatycznych Wstęp do kryptologii*”, WAT, Warszawa 2005
11. Ireneusz Kubiak, Artur Przybysz, „*Zdradziecki prąd*”, Chip 1/2005, str. 176
(http://www.chip.pl/arts/archiwum/n/articlear_120593.html)
12. Xiaoyun Wang i Hongbo Yu, „*How to Break MD5 and Other Hash Functions*”, Shandong University, Jinan 250100, luty 2005, Chiny
(<http://www.infosec.sdu.edu.cn/paper/md5-attack.pdf>)
13. Adam Rudziński, „*Fort Internet*”, Chip 2/2005, str. 142
(http://www.chip.pl/arts/archiwum/n/articlear_123015.html)
14. Bartosz Żółtak, „*Podpis do kosza ?*”, Chip 6/2005, str. 100
(http://www.chip.pl/arts/archiwum/n/articlear_134492.html)

15. Bartosz Żółtak, „Nie do złamania ?”, Chip 10/2005, str. 96
(http://www.chip.pl/arts/archiwum/n/articlear_144507.html)
16. Eryk Algo, „Pliki pod ochroną”, Chip 12/2005, str. 118
(http://www.chip.pl/arts/archiwum/n/articlear_157081.html)
17. Filip Zagórski, „Wyszukiwanie tajnych informacji w Google'u”, Chip 2/2006, str. 113
(http://www.chip.pl/arts/archiwum/n/articlear_165153.html)
18. Tomasz Trejderowski, „Włamanie do umysłu”, Chip 3/2006, str. 160
(http://www.chip.pl/arts/archiwum/n/articlear_167610.html)
19. Paweł Niedziejko, Ireneusz Krysowaty, „Biometria - Charakterystyka danych człowieka i ich wykorzystanie w bezpieczeństwie”, Zabezpieczenia, 4/2006
(http://www.zabezpieczenia.com.pl/archiwum/4_2006/kontrola-dostepu-biometria.html)
20. Paweł Niedziejko, Ireneusz Krysowaty, „Biometria - Charakterystyka danych człowieka i ich wykorzystanie w bezpieczeństwie (cd.)”, Zabezpieczenia, 5/2006
(http://www.zabezpieczenia.com.pl/archiwum/5_2006/kontrola-dostepu-biometria.pdf)
21. Paweł Niedziejko, Ireneusz Krysowaty, „Biometria - Charakterystyka danych człowieka i ich wykorzystanie w bezpieczeństwie (cd.)”, Zabezpieczenia, 6/2006
(http://www.zabezpieczenia.com.pl/archiwum/6_2006/kontrola-dostepu-biometria.pdf)
22. Tomasz Borukało, „Bezbronne pecety”, Chip 6/2006, str. 80
(http://www.chip.pl/arts/archiwum/n/articlear_171966.html)
23. Jan Lukáš, Jessica Fridrich i Miroslav Goljan, „Digital Camera Identification from Sensor Pattern Noise”, czerwiec 2006
(<http://www.ws.binghamton.edu/fridrich/>)
24. Szymon Piłat, Kamil Kulesza, „Niebezpieczne pozostałości”, Chip 7/2006, str. 140
(http://www.chip.pl/arts/archiwum/n/articlear_174400.html)
25. artykuł nr 102 716 Bazy Wiedzy (Knowledge Base) firmy Microsoft „NTLM user authentication in Windows”, Microsoft, Redmond USA, 1 listopad 2006
(<http://support.microsoft.com/kb/102716/>)
26. Grzegorz Tworek, „Microsoft Windows Vista Bitlocker Drive Encryption – jak to ugryźć, cz. I” i „Microsoft Windows Vista Bitlocker Drive Encryption – jak to ugryźć, cz. II”, listopad 2006
(<http://www.microsoft.com/poland/technet/article/art021.msp> i <http://www.microsoft.com/poland/technet/article/art022.msp>)
27. Jerzy Stanik, „Zarządzanie wymaganiami”, WAT, Warszawa 2007
28. Krzysztof Pietrzak, „Nowy bank na nowy rok”, PC World Komputer 1/2007, str. 132 (artykułu brak w archiwum online
http://www.pcworld.pl/archiwum/numer_1566.html)
29. Paweł Brągoszewski, „Zostań hackerem w weekend”, PC World Komputer 4/2007, str. 86 (<http://www.pcworld.pl/artykuly/54527.html>)
30. Jan Daciuk, „Biometria”, Katedra Inżynierii Wiedzy Wydział ETI, Politechnika Gdańska
(http://www.eti.pg.gda.pl/katedry/kiw/dydaktyka/Multimedialne_Systemy_Inteaktywne/biometria4.pdf)

Publikacje NASK (Naukowej Akademickiej Sieci Komputerowej) można znaleźć na stronie <http://www.nask.pl/run/n/Publikacje>.

31. Adam Czajka, Andrzej Pacut, „*Twój PIN to TY, część I*” (Biuletyn NASK, styczeń-luty 2003, str. 21-26, 2003) i „*Twój PIN to TY, część II*” (Biuletyn NASK, marzec-kwiecień 2003, str. 18-24, 2003)
32. Adam Czajka, Andrzej Pacut, „*Biometryczne metody weryfikacji tożsamości*”, Biuletyn NASK, wrzesień-październik-listopad 2003, str. 14-16, 2003

Strona Markusa G. Kuhna jest umieszczona pod adresem <http://www.cl.cam.ac.uk/~mgk25/>

33. Oliver Kömmerling i Markus G. Kuhn „*Design Principles for Tamper-Resistant Smartcard Processors*”, 10-11 maj 1999
34. Markus G. Kuhn, „*Optical Time-Domain Eavesdropping Risks of CRT Displays*”, University of Cambridge, Computer Laboratory, Wielka Brytania, 2002
35. Markus G. Kuhn, „*Electromagnetic Eavesdropping Risks of Flat-Panel Displays*”, University of Cambridge, Computer Laboratory, Wielka Brytania, 2004

Akty prawne

36. Rozporządzenia Ministra Edukacji Narodowej i Sportu z dnia 18 lipca 2005 w sprawie dokumentacji przebiegu studiów - http://www.men.gov.pl/prawo/wszystkie/rozp_361.php
37. Statut Wojskowej Akademii Technicznej im. Jarosława Dąbrowskiego w Warszawie (załącznik do uchwały Senatu WAT Nr 40/II/2006 z dnia 23 lutego 2006 wraz ze zmianami wniesionymi uchwałą Senatu WAT nr 71/II/2006 z dnia 12 października 2006) – <http://www.wat.edu.pl/0001/statut.pdf>

Dzienniki Ustaw są dostępne w Internecie np. pod adresem <http://isip.sejm.gov.pl/prawo/index.html>.

38. Ustawa z dnia 4 lutego 1994 o prawie autorskim i prawach pokrewnych (Dziennik Ustaw z 1994 Nr 24 poz. 83)
39. Ustawa z dnia 6 czerwca 1997 – Kodeks Karny (Dziennik Ustaw z 1997 nr 88 poz. 553)
40. Ustawa z dnia 29 sierpnia 1997 o ochronie danych osobowych (tekst jednolity z Dziennika Ustaw z 2002 Nr 101, poz. 926 ze zmianami z dnia 22 stycznia 2004 z Dziennika Ustaw z 2004 Nr 33, poz. 285) - http://www.giodo.gov.pl/144/id_art/1700/
41. Ustawa z dnia 22 stycznia 1999 o ochronie informacji niejawnych (Dziennik Ustaw z 1999 nr 11 poz. 95)
42. Ustawa z dnia 18 września 2001 o podpisie elektronicznym (Dziennik Ustaw z 2001 Nr 130, poz. 1450)
43. Ustawa z dnia 5 lipca 2002 o ochronie niektórych usług świadczonych drogą elektroniczną opartych lub polegających na dostępie warunkowym (Dziennik Ustaw z 2002 Nr 126, poz. 1068)

Treść patentów USA dostępna jest np. pod adresem <http://www.uspto.gov/patft/>.

44. treść patentu USA nr 4 405 829 „*Cryptographic communications system and method*” złożonego 14 grudnia 1977 przez Rivest; Ronald L. (Belmont, MA), Shamir; Adi (Cambridge, MA), Adleman; Leonard M. (Arlington, MA) i przyznanego 20 września 1983 (patent jest związany z algorytmem RSA)
45. treść patentu USA nr 5 231 668 „*Digital signature algorithm*” złożonego 26 lipca 1991 przez Kravitz; David W. (Owings Mills, MD) i przyznanego 27 lipca 1993 (dotyczy DSA)
46. treść patentu USA nr 5 291 560 „*Biometric personal identification system based on iris analysis*” złożonego 15 lipca 1991 przez Daugman; John G. i przyznanego 1 marca 1994
47. treść patentu USA nr 6 247 813 „*Iris identification system and method of identifying a person through iris recognition*” złożonego 4 listopada 1999 przez Kim; Dae Hoon (Seoul, KR), Ryoo; Jang Soo (Daejun, KR) i przyznanego 19 czerwca 2001.

Prace samoistne

48. „*Sieci komputerowe*”, ISBN 83-204-0964-0, Wydawnictwo Naukowo-Techniczne, Warszawa 1988 (tłumaczenie dokonane przez Marian Suskiewicz, Janusz Piel, Waldemar Borkowski, Bogna Znojkiwicz-Ozyp z „*Computer Networks*” Andrew S. Tanenbaum)
49. „*Bezpieczny kod Tworzenie i zastosowanie*”, ISBN 83-88440-19-5, Microsoft Press / APN PROMISE Sp. z o.o., Warszawa 2002 (tłumaczenie dokonane przez Rafał Otocky z „*Writing Secure Code*” Michael Howard i David LeBlane)
50. Piotr Metzger, „*Anatomia PC*” wydanie IX, ISBN 83-7361-507-5, Wydawnictwo Helion, 2004
51. „*Podpis elektroniczny – sposób działania, zastosowanie i korzyści*”, ISBN 83-914536-4-2, Ministerstwo Gospodarki, Warszawa 2005
(http://www.mgip.gov.pl/NR/rdonlyres/FCD37B0F-6B7B-46FB-8BE0-5EAC2DE085C1/15625/211105_podpis_elektroniczny.pdf)
52. „*Teoria bezpieczeństwa systemów komputerowych*”, Wydawnictwo Helion, ISBN 83-7361-678-0 (tłumaczenie dokonane przez Andrzej Grażyński, Tomasz Żmijewski z „*Fundamentals of Computer Security*” Josef Pieprzyk, Thomas Hardjono, Jennifer Seberry)

Specyfikacje

Specyfikacje platformy TPM są dostępne w Internecie pod adresem <https://www.trustedcomputinggroup.org/specs/TPM>.

53. „*Trusted Computing Platform Alliance (TCPA) Main Specification Version 1.1b*”, Trusted Computing Group, 2003
54. „*TPM Main Part 1 Design Principles Specification Version 1.2 Revision 94*”, Trusted Computing Group, 29 marca 2006

Przedstawione poniżej dokumenty RFC (Requests For Comments) są dostępne w Internecie np. w wyszukiwarce <http://www.rfc-editor.org/>

55. „*RFC 1319 The MD2 Message-Digest Algorithm*”, B. Kaliski, kwiecień 1992
56. „*RFC 1320 The MD4 Message-Digest Algorithm*”, R. Rivest, kwiecień 1992

57. „RFC 1321 The MD5 Message-Digest Algorithm”, R. Rivest, kwiecień 1992
58. „RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1”, The Internet Society, czerwiec 1999
59. „RFC 3174 US Secure Hash Algorithm 1 (SHA1)”, D. Eastlake 3rd, P. Jones, wrzesień 2001
60. „RFC 4634 US Secure Hash Algorithms (SHA and HMAC-SHA)”, D. Eastlake 3rd, P. Jones, sierpień 2006

Dokumenty FIPS PUB (Federal Information Processing Standards Publication) zwane też czasami FIPS np. na stronie <http://www.itl.nist.gov/fipspubs/>.

61. „Federal Information Processing Standards Publication 46-3 DATA ENCRYPTION STANDARD (DES)”, U.S. Department of Commerce, National Institute of Standards and Technology (NIST), USA, 25 październik 1999
62. „Federal Information Processing Standards Publication 140-2 DATA SECURITY REQUIREMENTS FOR CRYPTOGRAPHIC MODULES”, National Institute of Standards and Technology (NIST), USA, 25 maj 2001
63. „Federal Information Processing Standards Publication 180-2 Specifications for the SECURE HASH STANDARD”, U.S. Department of Commerce, National Institute of Standards and Technology (NIST), Information Technology Laboratory (ITL), USA, 1 sierpień 2002
64. „Federal Information Processing Standards Publication 186-2 DIGITAL SIGNATURE STANDARD (DSS)”, U.S. Department of Commerce, National Institute of Standards and Technology (NIST), USA, 27 stycznia 2000
65. „Federal Information Processing Standards Publication 197 Specification for the ADVANCED ENCRYPTION STANDARD (AES)”, U.S. Department of Commerce, National Institute of Standards and Technology (NIST), Information Technology Laboratory (ITL), USA, 26 listopad 2001

Inne pozycje bibliograficzne

66. strony organizacji standaryzacyjnych
 - a. ISO (International Organization for Standardization) - <http://www.iso.org>
 - b. IEC (International Electrotechnical Commission) - <http://www.iec.ch>
67. specyfikacje i informacje producentów
 - a. czytniki biometryczne
 - i. firma Rayco Security - <http://www.raycosecurity.com/biometrics/EyeDentify.html>
 - ii. firma Atmel (FingerChip) - <http://www.atmel.com/products/Biometrics/>
 - iii. firma Authentec - <http://www.authentec.com/technology.cfm>
 - b. klucze sprzętowe
 - i. HASP - <http://www.aladdin.com/HASP/>
 - ii. DESKey - <http://www.deskey.co.uk/>
 - c. moduły kryptograficzne
 - i. CryptoCard multiSIGN firmy CryptoTech - http://www.cryptotech.com.pl/compl/data/cryptotech/pdf/CryptoCard_multiSign.pdf

- ii. eToken PRO firmy Aladdin - <http://www.aladdin.com/etoken/pro/usb.asp>
 - iii. układ TPM firmy Infineon - <http://www.infineon.com/cgi-bin/ifx/portal/ep/channelView.do?channelId=-84648&channelPage=%2Fep%2Fchannel%2FproductOverview.jsp&pageTypeId=17099>
 - d. Kody kreskowe 2D firmy Veritec Inc. - <http://www.veritecinc.com/>
68. słowniki i encyklopedie
- a. Wikipedia w wersji polskiej – <http://www.wikipedia.pl>
 - b. Wikipedia w wersji angielskiej – <http://www.wikipedia.com>
 - c. Internetowa Encyklopedia PWN - <http://encyklopedia.wp.pl>
 - d. Wielka Interaktywna Encyklopedia Multimedialna (WIEM) - <http://portalwiedzy.onet.pl/encyklopedia.html>
69. strony instytucji edukacyjnych i usług przez nie udostępnianych
- a. Wojskowa Akademia Techniczna
 - i. strona główna – <http://www.wat.edu.pl>
 - ii. Wydział Cybernetyki – <http://ww.wcy.wat.edu.pl>
 - iii. biblioteka studencka – <http://ww.bg.wat.edu.pl>
 - iv. MSDNAA - http://msdn60.e-academy.com/elms/Storefront/Home.aspx?campus=msdnaa_kv5609
 - b. Politechnika Warszawska
 - i. strona główna – <http://www.pw.edu.pl>
 - ii. Ośrodek Kształcenia na Odległość (OKNO) – <http://www.okno.pw.edu.pl>
 - c. Uniwersytet Warszawski
 - i. strona główna – <http://www.uw.edu.pl>
 - ii. Centrum Otwartej i Multimedialnej Edukacji (COME) - <http://www.come.uw.edu.pl>
 - d. Uniwersytet Jagielloński
 - i. strona główna – <http://www.uj.edu.pl>
 - ii. Elektroniczna Rejestracja Kandydatów - <https://www.erk.uj.edu.pl/>
 - e. akademia Cisco - <http://cisco.netacad.net>
70. strony banków internetowych
- a. BZ WBK - <http://dlaciebie.bzwbk.pl/>
 - b. mBank – <http://www.mbank.pl>
 - c. LUKAS Bank - <http://www.lukasbank.pl/>
 - d. BPH - <http://www.bph.pl>
71. strony domowe programów i pakietów oprogramowania
- a. przeglądarki www
 - i. Internet Explorer – <http://www.microsoft.com/ie>
 - ii. Mozilla FireFox – <http://www.mozilla.com>
 - iii. Opera – <http://www.opera.com>
 - b. producenci programów antywirusowych
 - i. Kaspersky Lab – <http://www.kaspersky.pl>
 - ii. Symantec – <http://www.symantec.pl>
 - c. bazy danych
 - i. MySQL – <http://www.mysql.org>
 - ii. PostgreSQL - <http://www.postgresql.org>

- d. projekt Moodle - <http://moodle.org>
- e. program Ophrack - <http://ophcrack.sourceforge.net/>
- f. Sokrates-Dziekanat - <http://www.cs.put.poznan.pl/sokrates/index.html>
- g. systemy biblioteczne SOWA - <http://www.sokrates.pl>
- h. pakiet Gammu do obsługi telefonów komórkowych –
<http://www.gammu.org>
- i. serwer http Apache – <http://www.apache.org>
- j. język skryptowy PHP – <http://www.php.net>
- k. phpMyAdmin - <http://www.phpmyadmin.net/>
- l. JAP - http://anon.inf.tu-dresden.de/index_en.html

72. inne

- a. producent kart chipowych Gemplus – <http://www.gemplus.com>
- b. RSA Security Inc. - <http://www.rsasecurity.com>
- c. strona Ronalda Linn Rivesta (twórcy MD2-MD5) -
<http://theory.lcs.mit.edu/~rivest/>
- d. strona Philippe Oechslina - <http://lasecwww.epfl.ch/~oechslin>
- e. Decryptum – <http://www.decryptum.com>
- f. „łamanie” DES -
http://www.eff.org/Privacy/Crypto/Crypto_misc/DESCracker/
- g. strona algorytmu Blowfish - <http://www.schneier.com/blowfish.html>
- h. sprzętowy keylogger Key Shark -
<http://www.gadgets.co.uk/item/KEYSHARK/Keyboard-Key-Logger.html>
- i. klawiatura ekranowa Transec -
<http://www.micromata.de/produkte/transec.jsp>
- j. „złamanie” klawiatury ekranowej CityBanku -
<http://www.tracingbug.com/index.php/articles/view/23.html>
- k. OCR Research Team - <http://www.ocr-research.org.ua/>
- l. strona operatora sieci GSM Era – <http://www.era.pl>
- m. specyfikacje GSM - <http://pda.etsi.org/pda/queryform.asp>
- n. teksty o DRM w systemie Microsoft Vista - oryginał
http://www.cs.auckland.ac.nz/~pgut001/pubs/vista_cost.txt lub polski
przekład http://byte.livenet.pl/?page_id=819
- o. serwis <http://hacking.pl>

Indeks

A	
A5/1.....	21
A5/2.....	21
AES (Advanced Encryption Standard)	
.....	22
B	
Blowfish.....	22
C	
CAPTCHA (Completely Automated	
Public Turing test to tell Computers	
and Humans Apart).....	41
certyfikat.....	31
CMS (Course Management System)	11
CRC (Cyclic Redundancy Check).....	19
CRL (Certificate Revocation List).....	32
D	
DES (Data Encryption Standard).....	21
DES Cracker.....	22
DESKey.....	53
DSA (Digital Signature Algorithm)...	24
E	
e-learning.....	10
EAN (European Article Number).....	51
EGM (Elastic Graph Matching).....	57
F	
FAR (False Acceptance Rate).....	16
FRR (False Rejection Rate).....	16
funkcja Feistela.....	21
funkcje „skrótów”.....	18
H	
hasła.....	
jednorazowe.....	45
wielokrotne.....	34
HASP.....	53
HIP (Human Interactive Proof).....	41
I	
ICC (Integrated Circuit(s) Cards)....	48
IDEA (International Data Encryption	
Algorithm).....	22
K	
karty.....	
stykowe z paskiem magnetycznym	
.....	46
stykowe z układami elektronicznymi	
.....	48
inteligentne (smart cards).....	48
kryptograficzne.....	48
SIM (Subscriber Identification	
Module).....	49
zblizeniowe.....	50
keylogger.....	33
klawiatury ekranowe.....	34
klucze sprzętowe (dongles).....	53
L	
linie papilarne (epidermal ridges)....	55
login.....	35
M	
MD2.....	19
MD4.....	19
MD5.....	19
O	
odciski palców (fingerprints).....	55
P	
PCA (Principal Component Analysis)	57
phishing.....	37, 47
PIN (Personal Identification Number)	
.....	17, 45, 47
PKCS (Public Key Cryptography	
Standards).....	49, 53
PKI (Public Key Infrastructure).....	31
R	
RFID (Radio Frequency IDentification)	
.....	51
Rijndael.....	22
RSA (Rivest Shamir Adleman).....	23
S	
SHA (Secure Hash Algorithm).....	
SHA-1.....	19
SHA-2.....	19
SHA-2.....	
SHA-224.....	19
SHA-256.....	19
SHA-384.....	19
SHA-512.....	19
skimming.....	47
sola (salt).....	20
SSL (Secure Socket Layer).....	30
szyfrowanie.....	
asymetryczne.....	22
symetryczne.....	20
T	
tęczowe tablice (rainbow tables)....	20
token.....	45, 52
TPM (Trusted Platform Module).....	52
Trusted Computing.....	53

Trusted Computing Group.....	52	wyzwanie-odpowiedź (challenge-	
U		response).....	26
uwierzytelnianie.....	12	X	
W		XOR.....	21
WEP (Wired Equivalent Privacy).....	27		

Spis rysunków

Rysunek 1. Grupy użytkowników związane z instytucją edukacyjną (źródło: opracowanie własne).....	7
Rysunek 2. Tendencją widoczną również w instytucjach edukacyjnych jest łączenie jak największej liczby systemów ze sobą i udostępnianie ich usług bezprzewodowo oraz w Internecie (źródło: opracowanie własne).....	8
Rysunek 3. W systemach informatycznych problemem jest przeprowadzenie kontroli tego, czy dane są tam wprowadzane samodzielnie przez użytkownika czy nie (źródło: opracowanie własne).....	9
Rysunek 4. Schemat działania jednokierunkowych funkcji skrótu (źródło: opracowanie własne).....	19
Rysunek 5. Schemat działania szyfrów symetrycznych (źródło: opracowanie własne).....	20
Rysunek 6. Schemat działania funkcji szyfrowania asymetrycznego (źródło: opracowanie własne).....	23
Rysunek 7. Najprostszy schemat uwierzytelniania wykorzystuje tylko porównywanie z danymi wzorcowymi (źródło: opracowanie własne).....	25
Rysunek 8. Schemat uwierzytelniania „wyzwanie-odpowiedź” (źródło: opracowanie własne).....	27
Rysunek 9. Wynik eksperymentu Kaspersky Lab sprawdzającego zabezpieczenia sieci WiFi (dane z Warszawy z kwietnia 2007, podobnie jest w innych krajach)...	28
Rysunek 10. Schemat uwierzytelniania wykorzystywany m.in. w podpisie elektronicznym (źródło: opracowanie własne).....	29
Rysunek 11. Najbardziej ogólny schemat uwierzytelniania używanego np. w SSL (źródło: opracowanie własne).....	30
Rysunek 12. Przykład danych certyfikatu w przeglądarce Mozilla Firefox.....	31
Rysunek 13. Wygląd przykładowej klawiatury Transec.....	34
Rysunek 14. Liczba możliwych do wygenerowania haseł w zależności od ich długości i wielkości alfabetu (źródło: opracowanie własne).....	36
Rysunek 15. Przykład emaila służącego wyłudzeniu.....	37
Rysunek 16. Opcje i ich domyślne wartości dla haseł kont użytkowników w Microsoft Windows XP Professional PL (źródło: okno „Narzędzi administracyjnych” w „Panelu Sterowania”).....	39
Rysunek 17. Przykład wykorzystania odpowiedzi na pytania w systemie poczty elektronicznej Wirtualnej Polski (służą wyłącznie do odzyskiwania hasła).....	40
Rysunek 18. Przykład obrazka typu captcha generowanego w formularzu do tworzenia konta poczty elektronicznej w portalu Gazeta.pl.....	41
Rysunek 19. Rozwiązanie typu captcha zaproponowane w rozszerzeniu ConfirmEdit do pakietu MediaWiki.....	41
Rysunek 20. Dane z eksperymentu opisanego w magazynie „Chip” pokazujące, jak użytkownicy usuwają dane z dysków twardych.....	42
Rysunek 21. Token LUKAS Banku (źródło: strona www LUKAS Banku).....	45
Rysunek 22. Przykład karty płatniczej z paskiem magnetycznym (źródło: galeria wypukłych kart płatniczych serwisu KartyOnline).....	47
Rysunek 23. Karta hybrydowa Visa Electron wydana przez Bank Zachodni WBK S.A. (źródło: galeria płaskich kart płatniczych serwisu KartyOnline).....	48
Rysunek 24. Karta SIM (źródło: serwis Jakuba Bakxa).....	49

Rysunek 25. Wzór chipowej Elektronicznej Legitymacji Studenckiej (źródło: załącznik nr 3 do Rozporządzenia Ministra Edukacji Narodowej i Sportu z dnia 18 lipca 2005 w sprawie dokumentacji przebiegu studiów).....	50
Rysunek 26. Tekst "Marcin Wiacek" zapisany w standardzie QuickMark.....	51
Rysunek 27. Przykład czytnika linii papilarnych.....	56
Rysunek 28. Przykład zdjęcia tęczówki oka.....	58
Rysunek 29. Czytnik tęczówki oka BM-ET330 firmy Panasonic.....	58
Rysunek 30. Przykład obrazu żył w dłoni.....	59
Rysunek 31. ICAM 2001.....	61
Rysunek 32. Rozwiązanie wykorzystywać będzie dwa praktycznie niezależne kanały do podawania danych identyfikacyjnych.....	65
Rysunek 33. Schemat uwierzytelniania projektowanego podsystemu (jest to modyfikacja schematu „wyzwanie-odpowieź”).....	66
Rysunek 34. Zarys architektury projektowanego podsystemu uwierzytelniania (kolorem zielonym oznaczono stworzone samodzielnie elementy).....	72
Rysunek 35. Struktura bazy danych.....	73
Rysunek 36. Schemat blokowy modułu uwierzytelniania.....	75
Rysunek 37. Schemat modułu do wyświetlania listy użytkowników.....	76
Rysunek 38. Schemat modułu do dodawania nowych użytkowników.....	77
Rysunek 39. Schemat środowiska testowego podsystemu uwierzytelniania.....	79
Rysunek 40. Wygląd zaimplementowanej strony do dodawania nowego użytkownika (przed wysłaniem danych).....	80
Rysunek 41. Wygląd modułu do wyświetlania listy użytkowników.....	81
Rysunek 42. Wygląd zaimplementowanej strony www z modułem uwierzytelniania (przed zalogowaniem się użytkownika).....	81
Rysunek 43. Informacja pokazywana użytkownikowi po poprawnym uwierzytelnieniu.....	82

Załącznik A

Poniżej umieszczono kody źródłowe plików składających się na projekt przedstawiony w rozdziale 3.

Plik loguj.php

```
<?php
include ('inne\wspolne.php');

napisz_naglowki();
?>
<html>
<head>
  <title>Logowanie użytkownika</title>
  <meta name="author" content="Marcin Wiącek">
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
<?php
if (isset($_POST['C']) && isset($_POST['login2'])) {
} else {
  echo " <meta http-equiv='refresh' content='118' />\n";
}
?>
</head>
<body>
<?php
napisz_style();

//łączenie się z bazą danych
$blad = false;
$dbpass2 = @mysql_connect("$dbip","$dbuser","$dbpass");
if (!$dbpass) $blad = true;
if (!$blad) $dbconnect = @mysql_select_db("$dbname2",$dbpass2);
if (!$dbconnect) $blad = true;
if (!$blad) $dbconnect = @mysql_select_db("$dbname1",$dbpass2);
if (!$dbconnect) $blad = true;
if ($blad) {
  echo "Problemy z połączeniem z bazą.";
  echo "</body></html>";
  exit;
}

//usunięcie sms
$sql = SQL('delete from inbox where UpdatedInDB<(NOW() - INTERVAL 122 SECOND)');
mysql_query($sql);

//usunięcie tych sesji logowania, którym upłynęła ważność
mysql_select_db("$dbname2",$dbpass2);
$sql = SQL('delete from temp where Expiration<now()');
mysql_query($sql);

if (isset($_POST['login3']) && isset($_POST['success3'])) {
  napisz_naglowek("Logowanie użytkownika");

  echo " <center><font class=tresc>Witamy w systemie";

  $success = $_POST['success3'];

  //pokazujemy datę i godzinę ostatniego logowania
  if ($success!= "") echo " <br>Twoje ostatnie logowanie ".$success;

  echo " </font></center>\n";

  //uaktualniamy m.in. datę i godzinę ostatniego logowania
  $sql = SQL("update users set Expiration=(NOW() + INTERVAL 5 SECOND) where Login = %s",$_POST['login3']);
  mysql_query($sql);

  echo "<script>\n";
  echo " document.writeln(\" <form method=POST name=login3>\");\n";
  echo " document.writeln(\" <input type=hidden name=login3 value=\".$_POST['login3'].>\");\n";
```

```

        echo " document.writeln("<input type=hidden name=success3 value='$success'>");\n";
        echo " document.writeln("</form>");\n";
        echo " //za 3 sekund strona zostanie przeładowana\n";
        echo " setTimeout(\"login3.submit();\",3*1000);\n";
        echo "</script>\n";

        echo "</body></html>";
        exit;
    }

    if (isset($_POST['C']) && isset($_POST['login2'])) {
        napisz_naglowek("Logowanie użytkownika");

        $blad = true;

        $sql = SQL('select * from temp where C=%s;', $_POST['C']);
        $result = mysql_query($sql);
        if (mysql_affected_rows() != 1) {
            sleep(20);
            echo " <center><font class=tresc>Błąd uwierzytelnienia</font>\n";
            echo " <p><a class=tresc href=loguj.php>Spróbuj ponownie</a></center>\n";
            echo "</body></html>";

            //unieważnienie sesji logowania
            $sql = SQL('delete from temp where C=%s;', $_POST['C']);
            mysql_query($sql);

            exit;
        }
        $rekord = mysql_fetch_array($result, MYSQL_ASSOC);

        for ($i=0;$i<$timeout;$i++) {
            $login = "";

            $sql = SQL('select * from users');
            $result2 = mysql_query($sql);
            while ($rekord2 = mysql_fetch_array($result2,MYSQL_ASSOC)) {
                $tmp = md5($rekord2["Login"].$rekord["A"]);
                if ($_POST['login2']!= $tmp) continue;

                //przygotowanie poprawnej treści SMS
                $tmp3 = "";
                for ($j=0;$j<3;$j++) {
                    $tmp3 = $tmp3.substr($rekord2["Password"],substr($rekord["Numbers"],$j,1)-1,1);
                }
                if ($rekord["D"] == 1) {
                    $tresc_sms = $tmp3.$rekord["B"];
                } else {
                    $tresc_sms = $rekord["B"].$tmp3;
                }

                mysql_select_db("$dbname1",$dbpass2);
                //sprawdzamy od razu kodowanie, udh i klasę
                $sql = SQL("select * from inbox where (Coding='Default_No_Compression' or
Coding='Unicode_No_Compression') and Class='-1' and UDH='')");
                $result3 = mysql_query($sql);
                while ($rekord3 = mysql_fetch_array($result3,MYSQL_ASSOC)) {
                    //sprawdzenie numeru nadawcy
                    $tmp2 = md5($rekord3["SenderNumber"].$rekord2["Password"]);
                    if ($tmp2 != $rekord2["PhoneNumberPassword"]) continue;

                    //sprawdzenie tresci sms
                    if ($rekord3["TextDecoded"] != $tresc_sms) continue;

                    //sprawdzenie smsc
                    mysql_select_db("$dbname2",$dbpass2);
                    $sql = SQL('select * from smsc where Number=%s',$rekord3["SMSCNumber"]);
                    $result4 = mysql_query($sql);
                    if (mysql_affected_rows() != 1) continue;

                    if ($login != "") $blad=true;

                    $login = $rekord2["Login"];
                    $success = $rekord2["Success"];
                    $blad = false;
                }
            }
        }
    }

```

```

        break;
    }
    mysql_select_db("$dbname2",$dbpass2);

    if ($blad) sleep(1);
}

if ($blad) {
    echo " <center><font class=tresc>Błąd uwierzytelnienia</font>\n";
    echo " <p><a class=tresc href=loguj.php>Spróbuj ponownie</a></center>\n";
    echo "</body></html>";

    //unieważnienie sesji logowania
    $sql = SQL('delete from temp where C=%s;', $_POST['C']);
    mysql_query($sql);

    exit;
}

//unieważnienie sesji logowania
$sql = SQL('delete from temp where C=%s;', $_POST['C']);
mysql_query($sql);

echo " <center><font class=tresc>Witamy w systemie";

//pokazujemy datę i godzinę ostatniego logowania
if ($success!= "") echo " <br>Twoje ostatnie logowanie ".$success;

echo " </font></center>\n";

//uaktualniamy m.in. datę i godzinę ostatniego logowania
$sql = SQL("update users set Success=NOW(),Expiration=(NOW() + INTERVAL 5 SECOND) where Login =
%s",$login);
mysql_query($sql);

echo "<script>\n";
echo " document.writeln(\" <form method=POST name=login3>\");\n";
echo " document.writeln(\" <input type=hidden name=login3 value=$login>\");\n";
echo " document.writeln(\" <input type=hidden name=success3 value='$success'>\");\n";
echo " document.writeln(\" </form>\");\n";
echo " //za 3 sekund strona zostanie przeładowana\n";
echo " setTimeout(\"login3.submit()\",3*1000);\n";
echo "</script>\n";

echo "</body></html>";
exit;
}

$A = rand();
$B = rand();
$D = rand(1,2);
$liczby1 = rand(1,6);
$liczby2 = rand(1,6);
$liczby3 = rand(1,6);
$liczby_razem = $liczby1.$liczby2.$liczby3;
$liczby = "$liczby1, $liczby2 i $liczby3";
while (true) {
    //generowanie C, dopóki nie będzie unikalne
    $C = rand();
    $sql = SQL('INSERT INTO temp (Expiration,A,B,C,D,Numbers) VALUES ((NOW() + INTERVAL 120 SECOND), %s,
%s, %s, %s, %s);', $A,$B,$C,$D,$liczby_razem);
    mysql_db_query($dbname2,$sql);
    if (mysql_affected_rows() == 1) break;
}

echo "<script src=inne/md5.js></script>\n";
echo "<script>\n<!--\n";
echo "//za 118 sekund strona zostanie przeładowana\n";
echo "setTimeout(\"document.location.href='loguj.php'\",118*1000);\n";
echo "//funkcja wywoływana po naciśnięciu klawisza klawiatury ekranowej\n";
echo "function wpisz(znak) {\n";
echo " if (znak == "") {\n";
echo " //usuwamy znak\n";

```

```

echo " document.forms.login.login.value =
document.forms.login.value.substring(0,document.forms.login.value.length-1);\n";
echo " } else {\n";
echo " //dodajemy znak\n";
echo " document.forms.login.value = document.forms.login.value+ znak;\n";
echo " }\n";
echo " document.forms.login.Del.focus();\n";
echo "}\n";
echo "//funkcja wywoływana po wysłaniu formularza\n";
echo "function wyslij() {\n";
echo " if (document.forms.login.value=="") {\n";
echo " alert('Proszę wpisać login');\n";
echo " return false;\n";
echo " }\n";
echo " wartosc = MD5(document.forms.login.value);\n";
echo " wartosc = MD5(wartosc+"\$A");\n";
echo " document.forms.login.login2.value = wartosc;\n";
echo " document.forms.login.value = "";\n";
echo " document.getElementById('warstwa1').style.visibility = 'hidden';\n";
echo " document.getElementById('warstwa1').style.height = '1';\n";
echo " document.getElementById('warstwa2').style.visibility = 'visible';\n";
echo " return true;\n";
echo "}\n-->\n";
echo "</script>\n";

napisz_naglowek("Logowanie użytkownika");

echo " <center><table><tr><td>\n";
echo " &nbsp;<p>\n";
echo " <div id=warstwa1>\n";
echo " <script>\n";
echo " <!--\n";
echo " document.writeln("\ <font class=tresc>W celu zalogowania się należy w przeciągu 2 minut</font>");\n";
echo " document.writeln("\ <ol>");\n";
echo " document.writeln("\ <li><font class=tresc>wysłać z telefonu komórkowego SMS");\n";
echo " document.writeln("\ na numer $numer<br>o treści zawierającej");\n";
if ($D == 1) {
echo " document.writeln("\ $liczby cyfrę hasła oraz (bez spacji) liczbę $B</font></li>");\n";
} else {
echo " document.writeln("\ liczbę $B oraz (bez spacji) $liczby literę hasła</font></li>");\n";
}
echo " document.writeln("\ <li><font class=tresc>poczekać na otrzymanie raportu o doręczeniu
SMS</font></li>");\n";
echo " document.writeln("\ <li><font class=tresc>wpisać login korzystając z klawiatury ekranowej i wcisnąć przycisk
LOGUJ</font></li>");\n";
echo " document.writeln("\ </ol>");\n";

echo " document.writeln("\ &nbsp;<p>");\n";
echo " document.writeln("\ <center><form method=POST name=login onSubmit='return wyslij();>");\n";
echo " document.writeln("\ <input type=hidden name=login2>");\n";
echo " document.writeln("\ <input type=hidden name=C value='$C'>");\n";
echo " document.writeln("\ <center><font class=tresc>Login</font> <input name=login maxlength=20 readonly
type=password><p>");\n";
napisz_klawiatura();
echo " document.writeln("\ <p><input type=submit value=LOGUJ>");\n";
echo " document.writeln("\ </center>");\n";
echo " document.writeln("\ </form></center>");\n";
echo " -->\n";
echo " </script>\n";
echo " </div>\n";
echo " <div id=warstwa2 style='visibility:hidden;'><center><font class=tresc>Oczekiwanie na SMS z poprawną
treścią</font></center></div>\n";
echo " <script>\n";
echo " </script>\n";
echo " <noscript>\n";
echo " <center><font class=tresc>Z przykrością informujemy,\n";
echo " <br>że Twoja przeglądarka WWW nie interpretuje teraz JavaScript\n";
echo " <br>i przez to nie można wyświetlić tej strony</font></center>\n";
echo " </noscript>\n";
echo " </td></tr>\n";
echo " </table></center>";

?>
</body>
</html>

```


Plik dodaj.php

```

<?php
include ('inne\wspolne.php');

napisz_naglowki();
?>
<html>
<head>
<title>Dodanie nowego użytkownika</title>
<meta name="author" content="Marcin Wiącek">
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
</head>
<body>
<?php
napisz_style();

if (isset($_POST['gsm_pass']) && isset($_POST['login2']) && isset($_POST['password']) && isset($_POST['name'])) {
    //łączenie się z bazą danych
    $blad = false;
    $dbpass2 = @mysql_connect("$dbip","$dbuser","$dbpass");
    if (!$dbpass) $blad = true;
    if (!$blad) $dbconnect = @mysql_select_db("$dbname2",$dbpass2);
    if (!$dbconnect) $blad = true;
    if ($blad) {
        echo "Problemy z połączeniem z bazą.";
        echo "</body></html>";
        exit;
    }

    napisz_naglowek("Dodanie nowego użytkownika");

    $sql = SQL('INSERT INTO users (Login,Password,PhoneNumberPassword,Name) VALUES (%s, %s, %s, %s);',
$_POST['login2'], $_POST['password'], $_POST['gsm_pass'], $_POST['name']);
    mysql_query($sql);
    if (mysql_affected_rows() != 1) {
        echo "    <center><font class=tresc>Błąd - spróbuj innego loginu</font><br>";
    } else {
        echo "    <center><font class=tresc>Użytkownik został dodany pomyślnie</font><br>";
    }
    echo "    <p><a class=tresc href=dodaj.php>Dodaj nowego użytkownika</a></center><br>";
} else {
    echo "    <script src=inne/md5.js></script><br>";
    echo "    <script><br>";
    echo "    //funkcja wywoływana po naciśnięciu klawisza klawiatury ekranowej<br>";
    echo "    function wpisz(znak) {<br>";
    echo "        if (znak == ") {<br>";
    echo "            //usuwamy znak<br>";
    echo "            document.forms.login.login.value =
document.forms.login.login.value.substring(0,document.forms.login.login.value.length-1);<br>";
    echo "        } else {<br>";
    echo "            //dodajemy znak<br>";
    echo "            document.forms.login.login.value = document.forms.login.login.value+ znak;<br>";
    echo "        }<br>";
    echo "    }<br>";
    echo "    //funkcja wywoływana po wysłaniu formularza<br>";
    echo "    function wyslij() {<br>";
    echo "        if (document.forms.login.login.value=="") {<br>";
    echo "            alert('Proszę wpisać login');<br>";
    echo "            return false;<br>";
    echo "        }<br>";
    echo "        if (document.forms.login.name.value=="") {<br>";
    echo "            alert('Proszę wpisać imię i nazwisko');<br>";
    echo "            return false;<br>";
    echo "        }<br>";
    echo "        if (document.forms.login.password.value==" || document.forms.login.password2.value=="") {<br>";
    echo "            alert('Proszę wpisać hasło');<br>";
    echo "            return false;<br>";
    echo "        }<br>";
    echo "        if (document.forms.login.password.value!=document.forms.login.password2.value) {<br>";
    echo "            alert('Hasła muszą być takie same');<br>";
    echo "            return false;<br>";
    echo "        }<br>";
    echo "        if (document.forms.login.password.value.length<6) {<br>";

```

```

echo "    alert('Za krótkie hasło');\n";
echo "    return false;\n";
echo "  }\n";
echo "  if (document.forms.login.gsm.value=="") {\n";
echo "    alert('Proszę wpisać numer telefonu GSM');\n";
echo "    return false;\n";
echo "  }\n";
echo "  if (document.forms.login.gsm.value.length!=12 || document.forms.login.gsm.value.substr(0,3)!='+48')
{\n";
echo "    alert('Niepoprawny format numeru telefonu GSM');\n";
echo "    return false;\n";
echo "  }\n";
echo "  for (i=1;i<12;i++) {\n";
echo "    if (document.forms.login.gsm.value.substr(i,1)<'0' || document.forms.login.gsm.value.substr(i,1)>'9')
{\n";
echo "      alert('W numerze telefonu GSM mogą być tylko cyfry');\n";
echo "      return false;\n";
echo "    }\n";
echo "  }\n";
echo "  wartosc = MD5(document.forms.login.gsm.value+document.forms.login.password.value);\n";
echo "  document.forms.login.gsm_pass.value = wartosc;\n";
echo "  wartosc = MD5(document.forms.login.login.value);\n";
echo "  document.forms.login.login2.value = wartosc;\n";
echo "  document.forms.login.login.value = ";\n";
echo "  document.forms.login.gsm.value = ";\n";
echo "  return true;\n";
echo "  }\n";
echo " </script>\n";

napisz_naglowek("Dodanie nowego użytkownika");

echo " &nbsp;<p>\n";
echo " <center><script>\n";
echo "  document.writeln("\<font class=tresc>Proszę wpisać login korzystając z klawiatury
ekranowej<br>");\n";
echo "  document.writeln("\oraz resztę poniższych danych z klawiatury komputera<br>");\n";
echo "  document.writeln("\i nacisnąć przycisk Dodaj<font><br>&nbsp;<br>");\n";
echo "  document.writeln("\<form method='POST' name=login onSubmit='return wyslij();'>");\n";
echo "  document.writeln("\  <input type=hidden name=login2>");\n";
echo "  document.writeln("\  <input type=hidden name=gsm_pass>");\n";
echo "  document.writeln("\  <table>");\n";
echo "  document.writeln("\    <tr>");\n";
echo "  document.writeln("\      <td><font class=tresc>Login </font></td>");\n";
echo "  document.writeln("\      <td><input name=login maxlength=20 readonly></td>");\n";
echo "  document.writeln("\    </tr>");\n";
echo "  document.writeln("\    <tr><td colspan=2 align=center>");\n";
napisz_klawiatura();
echo "  document.writeln("\      </td></tr>");\n";
echo "  document.writeln("\    <tr>");\n";
echo "  document.writeln("\      <td><font class=tresc>Imię i nazwisko </font></td>");\n";
echo "  document.writeln("\      <td><input name=name></td>");\n";
echo "  document.writeln("\    </tr>");\n";
echo "  document.writeln("\    <tr>");\n";
echo "  document.writeln("\      <td><font class=tresc>Hasło </font></td>");\n";
echo "  document.writeln("\      <td><input name=password maxlength=20></td>");\n";
echo "  document.writeln("\    </tr>");\n";
echo "  document.writeln("\    <tr>");\n";
echo "  document.writeln("\      <td><font class=tresc>Powtórzenie hasła </font></td>");\n";
echo "  document.writeln("\      <td><input name=password2 maxlength=20></td>");\n";
echo "  document.writeln("\    </tr>");\n";
echo "  document.writeln("\    <tr>");\n";
echo "  document.writeln("\      <td><font class=tresc>Numer telefonu GSM<br>(format
+48123456789)</font></td>");\n";
echo "  document.writeln("\      <td><input name=gsm maxlength=20></td>");\n";
echo "  document.writeln("\    </tr>");\n";
echo "  document.writeln("\    <tr>");\n";
echo "  document.writeln("\      <td colspan=2 align=center><input type=submit name=send
value=DODAJ></td>");\n";
echo "  document.writeln("\    </tr>");\n";
echo "  document.writeln("\  </table>");\n";
echo "  document.writeln("\</form>");\n";
echo " </script>\n";
echo " <noscript>\n";
echo "  <font class=tresc>Z przykrością informujemy,\n";
echo "  <br>że Twoja przeglądarka WWW nie interpretuje teraz JavaScript\n";

```

```

        echo "    <br>i przez to nie można wyświetlić tej strony</font>\n";
        echo " </noscript>\n";
        echo " </center>\n";
    }
?>
</body>
</html>

```

Plik lista.php

```

<?php
include ('inne\wspolne.php');

napisz_naglowki();
?>
<html>
<head>
    <title>Lista użytkowników systemu</title>
    <meta name="author" content="Marcin Wiącek">
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
    <meta http-equiv="refresh" content="4">
</head>
<body>
<?php
napisz_style();

//łączenie się z bazą danych
$blad = false;
$dbpass2 = @mysql_connect("$dbip","$dbuser","$dbpass");
if (!$dbpass) $blad = true;
if (!$blad) $dbconnect = @mysql_select_db("$dbname2",$dbpass2);
if (!$dbconnect) $blad = true;
if ($blad) {
    echo "Problemy z połączeniem z bazą.";
    echo "</body></html>";
    exit;
}

napisz_naglowek("Lista zarejestrowanych użytkowników");

echo " &nbsp;<p>\n";
echo " <center>\n <table border=1 cellspacing=0 cellpadding=0>\n";
echo " <tr>\n";
echo " <td align=center bgcolor=blue>\n";
echo " <font class=tresc color=white>&nbsp;&nbsp;&Imię i nazwisko użytkownika&nbsp;&nbsp;&</font>\n";
echo " </td>\n";
echo " <td align=center bgcolor=blue>\n";
echo " <font class=tresc color=white>&nbsp;&nbsp;&Data i godzina ostatniego logowania&nbsp;&nbsp;&</font>\n";
echo " </td>\n";
echo " <td align=center bgcolor=blue>\n";
echo " <font class=tresc color=white>&nbsp;&nbsp;&Zalogowany ?&nbsp;&nbsp;&</font>\n";
echo " </td>\n";
echo " </tr>\n";

$sql = SQL('SELECT Name,Success from users where Expiration>NOW()');
$result = mysql_query($sql);
while ($rekord = mysql_fetch_array($result,MYSQL_ASSOC)) {
    echo " <tr>\n";
    echo " <td align=center>\n";
    echo " <font class=tresc><b>".$rekord["Name"]."</b></font>\n";
    echo " </td>\n";
    echo " <td align=center>\n";
    echo " <font class=tresc><b>";
    if ($rekord["Success"] == " ") {
        echo "nigdy";
    } else {
        echo $rekord["Success"];
    }
    echo "</b></font>\n";
    echo " </td>\n";
    echo " <td align=center>\n";
    echo " <font class=tresc><b>&nbsp;&TAK&nbsp;&</b></font>\n";
    echo " </td>\n";
    echo " </tr>\n";
}

```

```

}
$sql = SQL('SELECT Name,Success from users where Expiration<=NOW() or Expiration IS NULL');
$result = mysql_query($sql);
while ($rekord = mysql_fetch_array($result,MYSQL_ASSOC)) {
    echo "    <tr>\n";
    echo "        <td align=center>\n";
    echo "            <font class=tresc color=gray>".$rekord["Name"]."</font>\n";
    echo "        </td>\n";
    echo "    <td align=center>\n";
    echo "        <font class=tresc color=gray>";
    if ($rekord["Success"] == "") {
        echo "nigdy";
    } else {
        echo $rekord["Success"];
    }
    echo "</font>\n";
    echo "    </td>\n";
    echo "        <td align=center>\n";
    echo "            <font class=tresc color=gray>&nbsp;NIE&nbsp;</font>\n";
    echo "        </td>\n";
    echo "    </tr>\n";
}
echo " </table>\n </center>\n";
?>
</body>
</html>

```

Plik wspolne.php

```

<?php

// ----- parametry do zmiany -----

// dane logowania do serwera MySQL
$dbip = "127.0.0.1"; // adres serwera
$dbuser = "root"; // użytkownik
$dbpass = "root"; // hasło użytkownika
// nazwa baz danych
$dbname1 = "mgr"; // baza z tabelami Gammu
$dbname2 = "mgr2"; // baza z tabelami systemu uwierzytelniania
// inne
$numer = "+48123456789"; // numer telefonu odbierającego SMS
$timeout = 20; // ile sekund strona logowania ma czekać na SMS

// ----- kod -----

// funkcja wypisuje jedną linijkę klawiatury ekranowej
function napisz_linie($klawisze) {
    for ($i=0;$i<strlen($klawisze);$i++) {
        echo "        document.write(\"<input OnClick=wpisz(String.fromCharCode(";
        echo ord($klawisze[$i]);
        echo ")); type=button value=\"";
        if ($klawisze[$i] == "\\") {
            echo "\\\\";
        } else if ($klawisze[$i] == "") {
            echo "&#39;";
        } else {
            echo $klawisze[$i];
        }
        echo "\">\");\n";
    }
}

// funkcja wypisuje klawiaturę ekranową
function napisz_klawiatura() {
    echo "        document.write(\"<nobr>\");\n";
    napisz_linie("`1234567890-=");
    echo "        document.write(\"<input OnClick=wpisz(\"; name=Del type=button value='Del'>\");\n";
    echo "        document.write(\"</nobr><br>\");\n";
    napisz_linie("qwertyuiop[]");
    echo "        document.writeln(\"<br>\");\n";
    napisz_linie("asdfghjkl;");
    echo "        document.writeln(\"<br>\");\n";
    napisz_linie("zxcvbnm,./\");
}

```

```

        echo "        document.writeln("<br>");\n";
    }

//funkcja wypisuje nagłówki, które próbują wymusić na przeglądarce www
//nie przechowywanie strony w swojej pamięci podręcznej
//musi być wywołana przed napisaniem <html>
//opracowano na podstawie http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html
function napisz_naglowki() {
    header("Cache-Control: no-store, no-cache, must-revalidate");
    header("Expires: Sun, 21 Dec 1980 14:30:00 GMT");
    header("Pragma: no-cache");
    header("Last-Modified: ".gmdate("D, d M Y H:i:s") . " GMT");
}

//funkcja wypisuje style CSS strony
function napisz_style() {
    echo " <style type='text/css'>\n";
    echo " font.tytul1 {font-size: 24pt;font-family: Tahoma; text-decoration: none}\n";
    echo " font.tytul2 {font-size: 18pt;font-family: Tahoma; text-decoration: none}\n";
    echo " font.tresc {font-size: 14pt;font-family: Tahoma; text-decoration: none; text-align: justify;}\n";
    echo " a.tresc {font-size: 14pt;font-family: Tahoma; text-decoration: none; text-align: justify;}\n";
    echo " </style>\n";
}

//funkcja wypisuje lewą część strony (strona tytułowa)
function napisz_naglowek($text) {
    echo " <center>\n <table>\n";
    echo " <tr>\n";
    echo " <td align=center valign=center>\n";
    echo " <img src=inne/logo.png width=120>\n";
    echo " </td>\n";
    echo " <td align=center valign=center>\n";
    echo " <font class=tytul1>WIRTUALNY DZIEKANAT</font><br>\n";
    echo " <font class=tytul2>$text</font>\n";
    echo " </td>\n";
    echo " </tr>\n";
    echo " </table>\n </center>\n";
}

//dwie funkcje służące do zapobiegania atakom typu SQL Injection
//źródło całego poniższego kodu i opis na http://hacking.pl/5845

$arrArguments = array();
$intArgumentIndex = 0;

function parseArgument($arrMatches) {
    global $arrArguments, $intArgumentIndex;

    $strMatch = $arrMatches[0];
    $strArgument = @$arrArguments[$intArgumentIndex++];

    switch ($strMatch) {
        case '%d': return (int)$strArgument;
        case '%s': return "".mysql_real_escape_string($strArgument)."";
        case '%b': return (int)((bool)$strArgument);
    }
}

function SQL($strSql) {
    global $arrArguments, $intArgumentIndex;

    $arrArgs = func_get_args();
    array_shift($arrArgs);
    $arrArguments = $arrArgs;
    $intArgumentIndex = 0;
    return preg_replace_callback('/(%[dsb])/', 'parseArgument', $strSql);
}

?>

```

Plik mgr2.sql

```

-- phpMyAdmin SQL Dump
--

```

```
-- Database: `mgr2`
--
-----

--
-- Table structure for table `smc`
--

CREATE TABLE `smc` (
  `Number` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Dumping data for table `smc`
--

INSERT INTO `smc` (`Number`) VALUES
('+48602951111');

-----

--
-- Table structure for table `temp`
--

CREATE TABLE `temp` (
  `Expiration` datetime NOT NULL,
  `A` bigint(20) NOT NULL,
  `B` bigint(20) NOT NULL,
  `C` bigint(20) NOT NULL,
  `D` smallint(6) NOT NULL,
  `Numbers` int(11) NOT NULL,
  UNIQUE KEY `INDEX_C` (`C`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----

--
-- Table structure for table `users`
--

CREATE TABLE `users` (
  `Login` text NOT NULL,
  `Password` text NOT NULL,
  `PhoneNumberPassword` text NOT NULL,
  `Success` datetime default NULL,
  `Name` text,
  `Expiration` datetime default NULL,
  UNIQUE KEY `Login_Index` (`Login`(6))
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Plik md5.js

```
/*
 * md5.js 1.0b 27/06/96
 *
 * Javascript implementation of the RSA Data Security, Inc. MD5
 * Message-Digest Algorithm.
 *
 * Copyright (c) 1996 Henri Torgemane. All Rights Reserved.
 *
 * Permission to use, copy, modify, and distribute this software
 * and its documentation for any purposes and without
 * fee is hereby granted provided that this copyright notice
 * appears in all copies.
 *
 * Of course, this soft is provided "as is" without express or implied
 * warranty of any kind.
 */
```

```
function array(n) {
```

```

for(i=0;i<n;i++) this[i]=0;
this.length=n;
}

/* Some basic logical functions had to be rewritten because of a bug in
 * Javascript.. Just try to compute 0xffffffff >> 4 with it..
 * Of course, these functions are slower than the original would be, but
 * at least, they work!
 */

function integer(n) { return n%(0xffffffff+1); }

function shr(a,b) {
  a=integer(a);
  b=integer(b);
  if (a-0x80000000>=0) {
    a=a%0x80000000;
    a>>=b;
    a+=0x40000000>>(b-1);
  } else
    a>>=b;
  return a;
}

function shl1(a) {
  a=a%0x80000000;
  if (a&0x40000000==0x40000000)
  {
    a-=0x40000000;
    a*=2;
    a+=0x80000000;
  } else
    a*=2;
  return a;
}

function shl(a,b) {
  a=integer(a);
  b=integer(b);
  for (var i=0;i<b;i++) a=shl1(a);
  return a;
}

function and(a,b) {
  a=integer(a);
  b=integer(b);
  var t1=(a-0x80000000);
  var t2=(b-0x80000000);
  if (t1>=0)
    if (t2>=0)
      return ((t1&t2)+0x80000000);
    else
      return (t1&b);
  else
    if (t2>=0)
      return (a&t2);
    else
      return (a&b);
}

function or(a,b) {
  a=integer(a);
  b=integer(b);
  var t1=(a-0x80000000);
  var t2=(b-0x80000000);
  if (t1>=0)
    if (t2>=0)
      return ((t1|t2)+0x80000000);
    else
      return ((t1|b)+0x80000000);
  else
    if (t2>=0)
      return ((a|t2)+0x80000000);
    else
      return (a|b);
}

```



```

}

function xor(a,b) {
  a=integer(a);
  b=integer(b);
  var t1=(a-0x80000000);
  var t2=(b-0x80000000);
  if (t1>=0)
    if (t2>=0)
      return (t1^t2);
    else
      return ((t1^b)+0x80000000);
  else
    if (t2>=0)
      return ((a^t2)+0x80000000);
    else
      return (a^b);
}

function not(a) {
  a=integer(a);
  return (0xffffffff-a);
}

/* Here begin the real algorithm */

var state = new array(4);
var count = new array(2);
  count[0] = 0;
  count[1] = 0;
var buffer = new array(64);
var transformBuffer = new array(16);
var digestBits = new array(16);

var S11 = 7;
var S12 = 12;
var S13 = 17;
var S14 = 22;
var S21 = 5;
var S22 = 9;
var S23 = 14;
var S24 = 20;
var S31 = 4;
var S32 = 11;
var S33 = 16;
var S34 = 23;
var S41 = 6;
var S42 = 10;
var S43 = 15;
var S44 = 21;

function F(x,y,z) {
  return or(and(x,y),and(not(x),z));
}

function G(x,y,z) {
  return or(and(x,z),and(y,not(z)));
}

function H(x,y,z) {
  return xor(xor(x,y),z);
}

function I(x,y,z) {
  return xor(y ,or(x , not(z)));
}

function rotateLeft(a,n) {
  return or(shl(a, n),(shr(a,(32 - n))));
}

function FF(a,b,c,d,x,s,ac) {
  a = a+F(b, c, d) + x + ac;
  a = rotateLeft(a, s);
  a = a+b;
}

```

```

    return a;
}

function GG(a,b,c,d,x,s,ac) {
    a = a+G(b, c, d) +x + ac;
    a = rotateLeft(a, s);
    a = a+b;
    return a;
}

function HH(a,b,c,d,x,s,ac) {
    a = a+H(b, c, d) + x + ac;
    a = rotateLeft(a, s);
    a = a+b;
    return a;
}

function II(a,b,c,d,x,s,ac) {
    a = a+I(b, c, d) + x + ac;
    a = rotateLeft(a, s);
    a = a+b;
    return a;
}

function transform(buf,offset) {
    var a=0, b=0, c=0, d=0;
    var x = transformBuffer;

    a = state[0];
    b = state[1];
    c = state[2];
    d = state[3];

    for (i = 0; i < 16; i++) {
        x[i] = and(buf[i*4+offset],0xff);
        for (j = 1; j < 4; j++) {
            x[i]+=shl(and(buf[i*4+j+offset] ,0xff), j * 8);
        }
    }

    /* Round 1 */
    a = FF ( a, b, c, d, x[ 0], S11, 0xd76aa478); /* 1 */
    d = FF ( d, a, b, c, x[ 1], S12, 0xe8c7b756); /* 2 */
    c = FF ( c, d, a, b, x[ 2], S13, 0x242070db); /* 3 */
    b = FF ( b, c, d, a, x[ 3], S14, 0xc1bdceee); /* 4 */
    a = FF ( a, b, c, d, x[ 4], S11, 0xf57c0faf); /* 5 */
    d = FF ( d, a, b, c, x[ 5], S12, 0x4787c62a); /* 6 */
    c = FF ( c, d, a, b, x[ 6], S13, 0xa8304613); /* 7 */
    b = FF ( b, c, d, a, x[ 7], S14, 0xfd469501); /* 8 */
    a = FF ( a, b, c, d, x[ 8], S11, 0x698098d8); /* 9 */
    d = FF ( d, a, b, c, x[ 9], S12, 0x8b44f7af); /* 10 */
    c = FF ( c, d, a, b, x[10], S13, 0xffff5bb1); /* 11 */
    b = FF ( b, c, d, a, x[11], S14, 0x895cd7be); /* 12 */
    a = FF ( a, b, c, d, x[12], S11, 0x6b901122); /* 13 */
    d = FF ( d, a, b, c, x[13], S12, 0xfd987193); /* 14 */
    c = FF ( c, d, a, b, x[14], S13, 0xa679438e); /* 15 */
    b = FF ( b, c, d, a, x[15], S14, 0x49b40821); /* 16 */

    /* Round 2 */
    a = GG ( a, b, c, d, x[ 1], S21, 0xf61e2562); /* 17 */
    d = GG ( d, a, b, c, x[ 6], S22, 0xc040b340); /* 18 */
    c = GG ( c, d, a, b, x[11], S23, 0x265e5a51); /* 19 */
    b = GG ( b, c, d, a, x[ 0], S24, 0xe9b6c7aa); /* 20 */
    a = GG ( a, b, c, d, x[ 5], S21, 0xd62f105d); /* 21 */
    d = GG ( d, a, b, c, x[10], S22, 0x2441453); /* 22 */
    c = GG ( c, d, a, b, x[15], S23, 0xd8a1e681); /* 23 */
    b = GG ( b, c, d, a, x[ 4], S24, 0xe7d3fbc8); /* 24 */
    a = GG ( a, b, c, d, x[ 9], S21, 0x21e1cde6); /* 25 */
    d = GG ( d, a, b, c, x[14], S22, 0xc33707d6); /* 26 */
    c = GG ( c, d, a, b, x[ 3], S23, 0xf4d50d87); /* 27 */
    b = GG ( b, c, d, a, x[ 8], S24, 0x455a14ed); /* 28 */
    a = GG ( a, b, c, d, x[13], S21, 0xa9e3e905); /* 29 */
    d = GG ( d, a, b, c, x[ 2], S22, 0xfcefa3f8); /* 30 */
    c = GG ( c, d, a, b, x[ 7], S23, 0x676f02d9); /* 31 */
    b = GG ( b, c, d, a, x[12], S24, 0x8d2a4c8a); /* 32 */

```

```

/* Round 3 */
a = HH ( a, b, c, d, x[ 5], S31, 0xfffa3942); /* 33 */
d = HH ( d, a, b, c, x[ 8], S32, 0x8771f681); /* 34 */
c = HH ( c, d, a, b, x[11], S33, 0x6d9d6122); /* 35 */
b = HH ( b, c, d, a, x[14], S34, 0xfde5380c); /* 36 */
a = HH ( a, b, c, d, x[ 1], S31, 0xa4beea44); /* 37 */
d = HH ( d, a, b, c, x[ 4], S32, 0x4bdecfa9); /* 38 */
c = HH ( c, d, a, b, x[ 7], S33, 0xf6bb4b60); /* 39 */
b = HH ( b, c, d, a, x[10], S34, 0xbebfb7c70); /* 40 */
a = HH ( a, b, c, d, x[13], S31, 0x289b7ec6); /* 41 */
d = HH ( d, a, b, c, x[ 0], S32, 0xeaa127fa); /* 42 */
c = HH ( c, d, a, b, x[ 3], S33, 0xd4ef3085); /* 43 */
b = HH ( b, c, d, a, x[ 6], S34, 0x4881d05); /* 44 */
a = HH ( a, b, c, d, x[ 9], S31, 0xd9d4d039); /* 45 */
d = HH ( d, a, b, c, x[12], S32, 0xe6db99e5); /* 46 */
c = HH ( c, d, a, b, x[15], S33, 0x1fa27cf8); /* 47 */
b = HH ( b, c, d, a, x[ 2], S34, 0xc4ac5665); /* 48 */

/* Round 4 */
a = II ( a, b, c, d, x[ 0], S41, 0xf4292244); /* 49 */
d = II ( d, a, b, c, x[ 7], S42, 0x432aff97); /* 50 */
c = II ( c, d, a, b, x[14], S43, 0xab9423a7); /* 51 */
b = II ( b, c, d, a, x[ 5], S44, 0xfc93a039); /* 52 */
a = II ( a, b, c, d, x[12], S41, 0x655b59c3); /* 53 */
d = II ( d, a, b, c, x[ 3], S42, 0x8f0ccc92); /* 54 */
c = II ( c, d, a, b, x[10], S43, 0xffeff47d); /* 55 */
b = II ( b, c, d, a, x[ 1], S44, 0x85845dd1); /* 56 */
a = II ( a, b, c, d, x[ 8], S41, 0x6fa87e4f); /* 57 */
d = II ( d, a, b, c, x[15], S42, 0xfe2ce6e0); /* 58 */
c = II ( c, d, a, b, x[ 6], S43, 0xa3014314); /* 59 */
b = II ( b, c, d, a, x[13], S44, 0x4e0811a1); /* 60 */
a = II ( a, b, c, d, x[ 4], S41, 0xf7537e82); /* 61 */
d = II ( d, a, b, c, x[11], S42, 0xbd3af235); /* 62 */
c = II ( c, d, a, b, x[ 2], S43, 0x2ad7d2bb); /* 63 */
b = II ( b, c, d, a, x[ 9], S44, 0xeb86d391); /* 64 */

state[0] +=a;
state[1] +=b;
state[2] +=c;
state[3] +=d;

}

function init() {
    count[0]=count[1] = 0;
    state[0] = 0x67452301;
    state[1] = 0xefcdab89;
    state[2] = 0x98badcfe;
    state[3] = 0x10325476;
    for (i = 0; i < digestBits.length; i++)
        digestBits[i] = 0;
}

function update(b) {
    var index,i;

    index = and(shr(count[0],3) , 0x3f);
    if (count[0]<0xffffffff-7)
        count[0] += 8;
    else {
        count[1]++;
        count[0]-=0xffffffff+1;
        count[0]+=8;
    }
    buffer[index] = and(b,0xff);
    if (index >= 63) {
        transform(buffer, 0);
    }
}

function finish() {
    var bits = new array(8);
    var padding;
    var i=0, index=0, padLen=0;

```

```

        for (i = 0; i < 4; i++) {
            bits[i] = and(shr(count[0],(i * 8)), 0xff);
        }
        for (i = 0; i < 4; i++) {
            bits[i+4]=and(shr(count[1],(i * 8)), 0xff);
        }
        index = and(shr(count[0], 3) ,0x3f);
        padLen = (index < 56) ? (56 - index) : (120 - index);
        padding = new array(64);
        padding[0] = 0x80;
        for (i=0;i<padLen;i++)
            update(padding[i]);
        for (i=0;i<8;i++)
            update(bits[i]);

        for (i = 0; i < 4; i++) {
            for (j = 0; j < 4; j++) {
                digestBits[i*4+j] = and(shr(state[i], (j * 8)) , 0xff);
            }
        }
    }
}

/* End of the MD5 algorithm */

function hexa(n) {
    var hexa_h = "0123456789abcdef";
    var hexa_c="";
    var hexa_m=n;
    for (hexa_i=0;hexa_i<8;hexa_i++) {
        hexa_c=hexa_h.charAt(Math.abs(hexa_m)%16)+hexa_c;
        hexa_m=Math.floor(hexa_m/16);
    }
    return hexa_c;
}

var ascii="01234567890123456789012345678901" +
    " !\"#$%&'()*+,-./0123456789;<=>?@ABCDEFGHIJKLMNopQRSTUVWXYZ"+
    "[\\]^_`abcdefghijklmnopqrstuvwxyz{|}~";

function MD5(entree)
{
    var l,s,k,ka,kb,kc,kd;

    init();
    for (k=0;k<entree.length;k++) {
        l=entree.charAt(k);
        update(ascii.lastIndexOf(l));
    }
    finish();
    ka=kb=kc=kd=0;
    for (i=0;i<4;i++) ka+=shl(digestBits[15-i], (i*8));
    for (i=4;i<8;i++) kb+=shl(digestBits[15-i], ((i-4)*8));
    for (i=8;i<12;i++) kc+=shl(digestBits[15-i], ((i-8)*8));
    for (i=12;i<16;i++) kd+=shl(digestBits[15-i], ((i-12)*8));
    s=hexa(kd)+hexa(kc)+hexa(kb)+hexa(ka);
    return s;
}

```

Oświadczenie

Wyrażam zgodę na udostępnienie mojej pracy przez Bibliotekę Główną WAT w czytelni oraz w ramach wypożyczeń międzybibliotecznych.

.....
Data

.....
Marcin Tomasz Wiącek

NOTKA BIBLIOGRAFICZNA Z PRACY DYPLOMOWEJ

Specjalność **INFORMATYCZNE SYSTEMY ZARZĄDZANIA**

**Projekt i realizacja podsystemu uwierzytelniania
zdalnych użytkowników systemu informatycznego
wspomagającego funkcjonowanie
instytucji edukacyjnej**

MGR – S

Dyplomant **MARCIN TOMASZ WIĄCEK, WCY**

INSTYTUT SYSTEMÓW INFORMATYCZNYCH

Kierownik dr inż. WIESŁAW BARCIKOWSKI

112 stron, 3 rozdziały, 7 tabel, 43 rysunki

TREŚĆ NOTKI

W pracy opisano wybrane potrzeby instytucji edukacyjnych (ze szczególnym naciskiem na wydziały akademickie) w zakresie stosowania systemów informatycznych z uwierzytelnieniem. Następnie przedstawiono rodzaje, wady i zalety metod uwierzytelniania stosowanych we współczesnych systemach informatycznych. Przygotowano również projekt i opartą na nim implementację podsystemu uwierzytelniania, którą można użyć w aplikacjach napisanych w języku PHP. Wykorzystuje ona hasła, loginy oraz wiadomości SMS.

.....
(data)

.....
(podpis kierownika pracy)




WOJSKOWA AKADEMIA TECHNICZNA WYDZIAŁ CYBERNETYKI



Projekt i realizacja podsystemu uwierzytelniania
zdalnych użytkowników systemu informatycznego
wspomagającego funkcjonowanie instytucji edukacyjnej

Logowanie użytkownika - Mozilla Firefox

Logowanie użytkownika

 **WIRTUALNY DZIEKANAT**
Logowanie użytkownika

W celu zalogowania się należy w przeciągu 2 minut


1. wysłać z telefonu komórkowego SMS na numer +48602123456 o treści zawierającej 5, 4 i 2 cyfrę hasła oraz (bez spacji) liczbę 30289
2. poczekać na otrzymanie raportu o doręczeniu SMS
3. wpisać login korzystając z klawiatury ekranowej i wcisnąć przycisk LOGUJ

Login



Done

Lista użytkowników systemu


 **WIRTUALNY DZIEKANAT**
Lista zarejestrowanych użytkowników

Imię i nazwisko użytkownika	Data i godzina ostatniego logowania
Grzegorz Brzęczyszczkiewicz	2007-06-10 18:06:22
Marcin Wiącek	2006-12-21 14:30:00
Jan Kowalski	nigdy

Done

Logowanie użytkownika - Mozilla Firefox

Logowanie użytkownika

 **WIRTUALNY DZIEKANAT**
Logowanie użytkownika

Witamy w systemie
ostatnie logowanie 2007-05-14 22:19:04

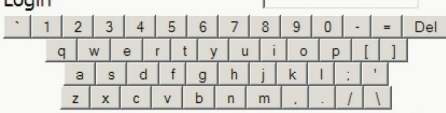
Logowanie użytkownika - Mozilla Firefox

Logowanie użytkownika

WIRTUALNY DZIEKANAT
Dodanie nowego użytkownika

Proszę wpisać login korzystając z klawiatury ekranowej oraz resztę poniższych danych z klawiatury komputera i nacisnąć przycisk Dodaj

Login



Imię i nazwisko

Hasło

Powtórzenie hasła

Numer telefonu GSM (format +48123456789)

Done

Dyplomant
Marcin Tomasz Wiącek

Kierownik pracy
dr inż. Wiesław Barcikowski