Złożenie pracy online:
2012-12-11 12:11:55
Kod pracy:
9063
Kod załącznika:
9027

Robert Podwika
(nr albumu: 10275*Z/SUM)

Praca magisterska

Bezpieczeństwo komunikacji internetowej na podstawie aplikacji komunikatora internetowego

Security of network communication upon internet messenger application.

Wydział: Nauk Społecznych i Informatyki

Kierunek: Zarządzanie

Specjalność: komputerowe wspomaganie decyzji i e-biznes

Promotor: dr inż. Bogdan Batko

Nowy Sącz, 2012

Zamierzeniem tej pracy jest przedstawienie zagadnień związanych z bezpieczeństwem komunikacji internetowej na praktycznym przykładzie własnego komunikatora internetowego. W pracy zostały omówione zagadnienia, które są niezbędne to prawidłowego zrozumienia tematu pracy takie jak: standardy bezpieczeństwa, istota oraz typy ataków komputerowych, kryptografia komunikacji sieciowej oraz konkurencyjne produkty, które mogą być substytutami aplikacji, która została napisana przeze mnie. Praca przedstawia również od początku budowę bezpiecznej aplikacji służącej komunikacji sieciowe oraz przedstawia wyniki raportu technicznego dotyczącego naruszeń bezpieczeństwa informacji w sieci.

The aim of this study is to present issues related to network security communication. This thesis covers topics that are essential to understand correctly the topic of thesis, such as the standards of security, the essence and types of computer attacks, the cryptography of network communication and the competitive products that could be the substitutes of the application written by me. The thesis also presents from the very beginning, building secure application aimed to network communication and discuss the results of technical research concerning violations of information security in network.

Spis treści

Wstęp

Szybka ewolucja technologii informacyjnych sprawiła, iż e-biznes zaczął się dynamicznie rozwijać. Obniżanie kosztów operacji i projektów, zwiększanie prędkości transakcji, globalny dostęp dla klientów i sprzedawców to powody druzgocącego wzrostu tej nowej drogi biznesu. Ponieważ duża liczba nakładów pieniężnych jest angażowana w e-biznesie, potrzebne są rozwiązania, które zapewnią bezpieczeństwo kapitału, transakcji oraz przesyłanych danych. Rola bezpieczeństwa nie jest wyolbrzymiona, ponieważ niskim nakładem kosztów możliwe jest pozyskanie cennych danych z firm, które nie są odpowiednio zabezpieczone. Po zbadaniu technologii wykorzystywanych w e-commerce należy zidentyfikować zagrożenia oraz stworzyć standardy bezpieczeństwa, a przede wszystkim dobre standardy kryptograficzne. Dobra firma powinna zabezpieczyć swój system przed wszelkiego rodzaju zagrożeniami oraz wyeliminować jego słabe punkty. Każda informacja przed wysłaniem jej do sieci jest przetwarzana jest komputer na kod binarny. Tenże kod jest enkapsulowany poprzez warstwy sieciowe modelu OSI[1] w zależności od wybranego protokołu komunikacyjnego. Kod zerojedynkowy zostaje następnie wysłany poprzez urządzenie zwane kartą sieciową do sieci. W zależności od architektury sieciowej transmisja może zostać podsłuchana. W przypadku niezaszyfrowanych danych osoba podsłuchująca może bardzo niskim nakładem sił otrzymać nasze hasła do serwisów internetowych, treść poczty e-mail lub też całe przesyłane pliku. W pracy przedstawię wygląd transmisji szyfrowanej przy użyciu programu, który napiszę oraz przykład, gdzie transmisja nie będzie szyfrowana.

W tej pracy postaram się przedstawić problem bezpieczeństwa komunikacji w firmach. Przedstawię podstawowe standardy bezpieczeństwa, które są niezbędne w nowoczesnej firmie działającej w Internecie. Spróbuję odpowiedzieć na pytanie;

[1] OSI (ang. Open System Interconnection) lub Model OSI (pełna nazwa ISO OSI RM, ang. ISO OSI Reference Model – model odniesienia łączenia systemów otwartych) – standard zdefiniowany przez ISO oraz ITU-T opisujący strukturę komunikacji sieciowej. Źródło: http://pl.wikipedia.org/wiki/Model_OSI (data odczytu 13.03.2012)

dlaczego firmy nie są świadome zagrożeń, które mogą się pojawić, gdy firma wkracza do Internetu. Celem pracy jest zaznajomienie czytelnika z podstawowymi problemami bezpieczeństwa, które pojawiają się podczas użytkowania komunikacji sieciowej oraz pokazanie na praktycznym przykładzie jak zabezpieczyć się przed nieautoryzowanym dostępem do informacji. W pracy zostaną przedstawione badania, które pokazują problem bezpieczeństwa komunikacji w dzisiejszych czasach zarówno w małych i średnich firmach, jak i w dużych korporacjach. Praca będzie zawierała informacje o typach ataków komputerowych oraz narzędzia, które służą do obrony przed nimi, począwszy od prostego szyfrowania, a kończąc na zaawansowanej kryptografii. Kolejnym z celów pracy jest przedstawienie alternatywnych aplikacji, które służą podniesieniu bezpieczeństwa informacji w sieci. Końcowym celem pracy jest napisanie aplikacji w Java, która zapewnia bezpieczeństwo komunikacji sieciowej i tworzy bezpieczny kanał komunikacyjny.

Potrzeba bezpieczeństwa informacji w e-biznesie

Pojęcie e-biznesu

Zanim zacznę opisywać potrzeby firm pracujących w biznesie elektronicznym, chciałbym przedstawić, czym jest e-biznes. E-biznes(ang. Electronic business) to wprowadzony w 1995 roku przez IBM według powszechnie obowiązującej definicji model prowadzenia biznesu opierający się na szeroko rozumianych rozwiązaniach teleinformatycznych w szczególności aplikacjach internetowych. Pojęcie elektronicznego biznesu obejmuje między innymi wymianę informacji między producentami, dystrybutorami i odbiorcami produktów i usług, zawieranie kontraktów, przesyłanie dokumentów, prowadzenie telekonferencji, pozyskiwanie nowych kontraktów, wyszukiwanie informacji et cetera.[2]

Historia e-biznesu

Historia biznesu elektronicznego ma swój początek w latach 60. XX wieku, kiedy to po raz pierwszy komputery zostały wykorzystane do celów komercyjnych. Dzięki informatyzacji banków poprzez wprowadzenie systemu ERMA(ang. Electronic Recording Machine Accounting) w Bank of America dziewięciu pracowników mogło wykonać pracę, którą wcześniej wykonywało pięćdziesięciu. Biznesowe zalety komputerów zostały bardzo szybko dostrzeżone przez firmy, które zajęły się automatyzacją procesów takich jak: administrowanie płacami, planowanie harmonogramów produkcji, prowadzenie księgowości, tworzenie raportów et cetera. Wprowadzenie informatyki znacznie przyspieszyło wykonywanie powyższych czynności. W latach 70. Oraz 80. XX wieku komputery zostały użyte do komunikacji pomiędzy firmami. Powstały pierwsze systemy do wymiany informacji EDI(ang. Electronic Data Interchange). W 1968 roku grupa przedsiębiorstw kolejowych sformułowała TDCC(ang. Transportation Data Coordinating Committee), który określał, iż protokół wymiany informacji musi być niezależny sprzętowo. Jego interfejs

[2] Źródło: http://pl.wikipedia.org/wiki/E-biznes (data odczytu: 13.03.2012)

powinien dostrzec potrzeby użytkownika. Powinien również zostawić wybór prędkości oraz usług, użytkownikowi. W 1985 TDCC obsługiwał szacunkowo 90% wszystkich listów przewozowych na kolei. Dzisiejsza wersja standardu zawiera:

- Eksport oraz import informacji dla przesyłek międzynarodowych,
- rezerwację oraz żądanie odbioru przesyłki,
- system informacji o statusie przesyłki,
- informacje o przesyłce od spedytora dla przewoźnika,
- dane dotyczące płatności.[3]

W dzisiejszych czasach istnieje wiele systemów do wymiany informacji. Najczęściej są to strony www(ang. World Wide Web), komunikatory internetowe, poczta e-mail(ang. Electronic Mail), serwery FTP(ang. File Transfer Protocol), komunikatory, systemy intranetowe[4] oraz ekstranetowe[5]. Firmy bardzo często wykorzystują gotowe dostępne oprogramowanie bez przyglądania się aspektowi bezpieczeństwa.

[3] Źródło: http://www.123edi.com/edi-tdcc-101.asp (data odczytu 13.03.2012)

[4] Intranet – sieć komputerowa ograniczająca się do komputerów w np. firmie lub organizacji. Źródło: http://pl.wikipedia.org/wiki/Intranet (data odczytu 13.03.2013)

[5] Ekstranet (ang. extranet) to rozwiązanie sieciowe polegające na połączeniu dwóch lub większej liczby intranetów za pomocą protokołów sieciowych. Źródło: http://pl.wikipedia.org/wiki/Ekstranet (data odczytu 13.03.2013)

Standardy bezpieczeństwa

Wzrost znaczenia e-biznesu w ostatnich latach sprawił, że potrzebne było stworzenie standardów w dziedzinie bezpieczeństwa informacji(zdefiniowanie, zaprojektowanie, implementacja oraz testowanie rozwiązań na systemach e-biznesowych). Praktyką w wielu firmach jest projektowanie rozwiązań opartych głownie na wiedzy i doświadczeniu indywidualnych osób wchodzących w skład grupy projektującej rozwiązanie. Aż do roku 2001 nie było żadnego opublikowanego standardu bezpieczeństwa.

Dokument „The common criteria"

W 1999 roku pierwszą próbą standaryzacji był dokument „The common criteria". Dokument sugerował, by poprzez ewaluację wykrywać błędy w oprogramowaniu mające istotne znaczenie dla bezpieczeństwa. Nie określał on jednak standardów projektowania systemów informatycznych i w całości opierał się na doświadczeniu indywidualnych pracowników. W tym przypadku każdego rodzaju defekt wykrywany jest bardzo późno, jeśli nawet dochodzi do jego odkrycia.

ISO/IEC 27001

Norma ISO/IEC 27001 jest międzynarodowym dokumentem standaryzującym systemy zarządzania bezpieczeństwem informacji. Została ogłoszona 14 października 2005 roku na podstawie brytyjskiego standardu BS 7799-2 opublikowanego przez BSI[6]. W Polsce norma ta została opublikowana 4 stycznia 2007 roku, jako PN-ISO/IEC 27001:2007. Norma ta zastąpiła PN-I-07799-2:2005 czyli polską wersję brytyjskiego standardu BS 7799-2.

Norma określa wymagania dla ustanowienia, wdrożenia, zarządzania, monitorowania i przeglądu udokumentowanego systemu zarządzania bezpieczeństwem informacji (SZBI, ang. ISMS – Information Security Management System) oraz 11 obszarów mechanizmów kontrolnych obejmujących:

[6] BS 7799 - brytyjski standard stanowiący podstawę systemów zarządzania bezpieczeństwem informacji opracowany przez BSI w 1995. Źródło: http://pl.wikipedia.org/wiki/BS_7799 (data odczytu 5.10.2012)

- politykę bezpieczeństwa,
- organizację bezpieczeństwa,
- zarządzanie aktywami,
- bezpieczeństwo zasobów ludzkich,
- bezpieczeństwo fizyczne i środowiskowe,
- zarządzanie systemami i sieciami,
- kontrolę dostępu,
- pozyskiwanie, rozwój i utrzymanie systemów informatycznych,
- zarządzanie incydentami bezpieczeństwa,
- zarządzanie ciągłością działania,
- zapewnienie zgodności z wymaganiami wewnętrznymi i prawnymi .

Powyższy standard gwarantuje w pełni kompleksowe podejście do problemu bezpieczeństwa informacji, obejmując swym zakresem nie tylko zagadnienia związane z teleinformatyką, lecz również z bezpieczeństwem osobowym, fizycznym oraz organizacyjno-prawnym.[7]

[7] Źródło: http://www.iso27000.pl/sites/view/site=85 (data odczytu 2.10.2012)

Raport techniczny dotyczący naruszenia bezpieczeństwa informacji

Poniżej przedstawię wyniki ostatniego raportu dotyczącego naruszenia bezpieczeństwa informacji.

Firmy zajmujące się przeprowadzeniem badania

Raporty ISBS są przeprowadzane, co kilka lat od początku 1990 roku przez firmy:

Rysunek 1. Logo firmy Infosecurity. Źródło: http://www.net-security.org/images/articles/InfoLogo_Europe-no-dates.gif(data odczytu 19.11.2012).

Jest to firma będąca 17 lat na rynku przemysłu IT. Organizuje ona spotkania, targi, imprezy informacyjne, które są bardzo ważne dla każdej osoby zajmującej się bezpieczeństwem informacji. Eventy, które firma organizuje odbywają się w wielu krajach między innymi: Belgi, Holandii oraz Rosji.

Rysunek 2. Logo firmy Reed Exhibitions. Źródło: http://1.bp.blogspot.com/_cxOwbSunBqc/TOuGtau0FI/AAAAAAAAAJ8/OcYnq0gA1H4/s1600/reed_exhibitions_logo.jpg(data odczytu 19.11.2012).

Reed Exhibitions jest jedną z największych firm organizujących ponad 500 spotkań w 39 krajach. W 2011 roku w spotkaniach z zakresu bezpieczeństwa uczestniczyło razem 6 milionów osób z całego świata, generujących biliony dolarów w biznesie. Aktualnie firma organizuje spotkania w Europie, Azji, na Bliskim Wschodzie w Afryce oraz obu Amerykach.

Raport jest napisany przez firmę:

Rysunek 3. Logo firmy Pwc. Źródło: http://www.cartell.ie/car_check/wp-content/uploads/2011/05/Pwc-logo-2010.jpg(data odczytu 19.11.2012).

która pomaga organizacjom i osobom prywatnym w zabezpieczeniu swoich informacji. Firma ta ma ponad 30 lat doświadczenia i zatrudnia ponad 200 specjalistów z zakresu bezpieczeństwa usług internetowych w Wielkiej Brytanii oraz ponad 3500 na całym świecie. Pwc zyskała międzynarodowe uznanie dla wiedzy technicznej związanej z strategią bezpieczeństwa informacji. Została uznana za lidera w branży bezpieczeństwa IT.

Informacje wstępne na temat raportu

Tegoroczny raport pokazuje, iż naruszenia bezpieczeństwa informacji są na historycznie wysokim poziomie i kosztują Wielką Brytanie miliardy funtów rocznie. W ciągu dwóch lat liczba ataków komputerowych się podwoiła. W raporcie zauważono także, firmy oszczędzają na bezpieczeństwie informacji i nie edukują personelu w zakresie ochrony danych. Wiele firm ma problemy z przeznaczaniem odpowiednich środków na bezpieczeństwo. Kluczowym działanie powinno być zbadanie korzyści biznesowych wynikających z inwestycji w bezpieczeństwo IT. W większości przypadków firmy ponoszą większe koszty wynikające z szkód spowodowanych przez włamania, aniżeli koszty, które zostałyby poniesione na prewencje i profilaktykę. Środowisko biznesowe się zmienia, nie jest stałe. Społeczności internetowe mają coraz większe znaczenie w biznesie. Firmy otwierają ich systemy na komórki, tablety, urządzenia mobilne. Informatycy muszą się zmierzyć z masą nowych technologii i często nie są w stanie zabezpieczyć wszystkiego. Większość respondentów oczekuję, iż liczba naruszeń bezpieczeństwa informacji zwiększy się w przyszłości.

Łącznie w badaniu przeprowadzonym w lutym oraz marcu 2012 roku uczestniczyło 447 organizacji. Liczba respondentów jest porównywalna z badaniem z 2010 roku. Margines błędu wynosi +/- 6% przy 95% poziomie ufności w grupie dużych firm oraz +/- 8% dla małych i średnich przedsiębiorstw. Osoby ankietowane pochodziły z różnych sektorów przemysłu. Najwięcej było osób, zajmujących się zarządzaniem biznesem, dyrektorów wykonawczych oraz osób, które zajmowały się bezpieczeństwem IT lub były członkami działu IT. Na rysunku poniżej znajduje się wykres przedstawiający informacje na temat branży działania respondentów.

Rysunek 4. Branże działania respondentów badania ISBS 2012. Źródło: opracowanie własne.

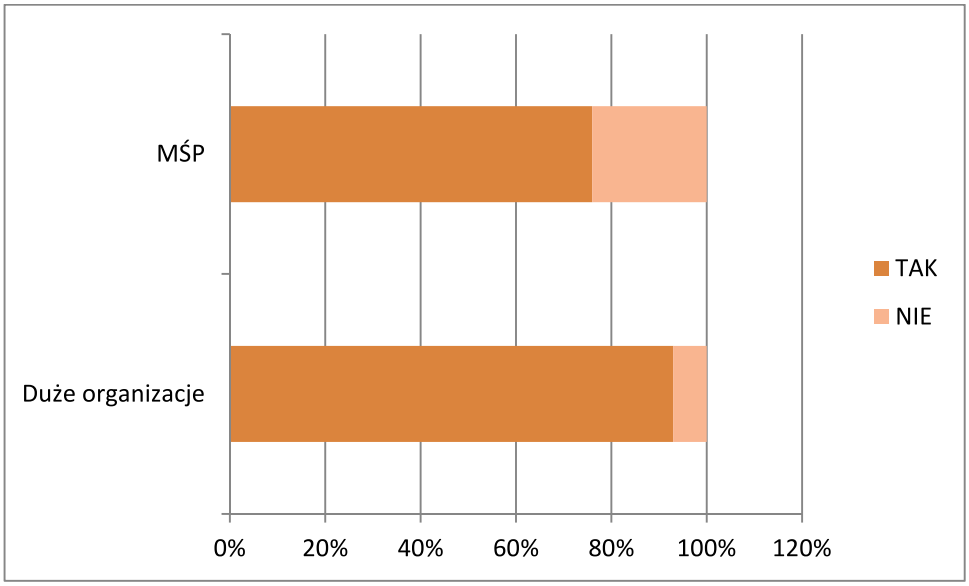
Na podstawie danych z raportu „Information security breaches survey 2012”

W jakim sektorze działa Pańska firma?		
Usługi finansowe	17%	17
Telekomunikacja	5%	5
Technologie	22%	22
Podróże, wypoczynek, rozrywka	2%	2
Narzędzia, górnictwo, energetyka	2%	2
Produkcja	6%	6
Handel i dystrybucja	3%	3
Nieruchomości i budownictwo	2%	2
Rząd, zdrowie lub edukacja	21%	21
Inne	20%	20
Razem	100%	100

Tabela 1. Wyniki ankiety pytającej klientów o branże działania ich firm. Źródło: opracowanie własne.

Streszczenie głównych informacji raportu

Według raportu zdecydowana większość respondentów miała do czynienia z naruszeniem bezpieczeństwa informacji w ciągu ostatniego roku.



Rysunek 5. Odpowiedzi respondentów na pytanie dotyczące naruszenia bezpieczeństwa w ich firmie w ostatnim roku. Źródło: opracowanie własne.

Mediana liczby istotnych ataków przez nieuprawnionych użytkowników w każdej dużej organizacji wyniosła 54. Jest to dwa razy więcej niż w roku 2010. 15% firm z sektora MŚP zostało zaatakowane DOS-em. 15% wielkich firm wykryło, że hakerzy włamali się do ich sieci w ostatnim roku.

Najpoważniejsze naruszenia bezpieczeństwa zostały spowodowane zaniedbaniami ludzi, procesów oraz technologii. Oszustwa komputerowe, utrata danych oraz naruszenia bezpieczeństwa informacji(wraz z atakami hackerskimi) najczęściej były bardzo poważne w skutkach. 45% dużych firm naruszyło politykę zabezpieczania danych(i stało się to, co najmniej raz dziennie w jednej firmie na dziesięć). 18% organizacji, które zostały ofiarami ataku miało przepisy o ochronie danych oraz skuteczny plan awaryjny. 19% dużych firm miało problemy przez pracowników, którzy przeprowadzali oszustwa komputerowe.

Większość przyczyn nieprawidłowości i ataków było powiązane z tym, że firmy inwestowały w edukację personelu po zaistniałym ataku. 44% dużych firm zorganizowało i zmieniło politykę oraz procedury bezpieczeństwa po zdiagnozowanym ataku na ich informacje. 26% organizacji, które posiadają politykę bezpieczeństwa wierzą, że ich personel bardzo dobrze rozumie zasady bezpieczeństwa. 75% z organizacji gdzie polityka bezpieczeństwa była słabo rozumiana, miało problemy z kwestiami bezpieczeństwa. 54% MŚP nie miało żadnego programu edukacji personelu związanego z bezpieczeństwem informacji.

W konsekwencji koszt naruszeń bezpieczeństwa informacji pozostaje bardzo wysoki. 15 000 – 30 000 funtów brytyjskich wyniósł średni koszt włamania w sektorze MŚP w Wielkiej Brytanii. 110 000 funtów brytyjskich wyniósł średni koszt naruszeń bezpieczeństwa informacji w dużych firmach. W bilionach funtów można przedstawić całkowity koszt naruszeń bezpieczeństwa informacji w całej Wielkiej Brytanii w ostatnim roku.

Audyty bezpieczeństwa nie nadążają z zmianami w biznesie. Internet stwarza możliwości do tworzenia bardziej wyrafinowanych relacji biznesowych. 73% respondentów posiada outsourcing procesów biznesowych poprzez Internet. 38% dużych organizacji zapewnia, że dane trzymane przez zewnętrznych dostawców są szyfrowane. 56% firm sektora MŚP nie dba o kontrolę bezpieczeństwa zewnętrznych firm dostarczających usługi bezpieczeństwa.

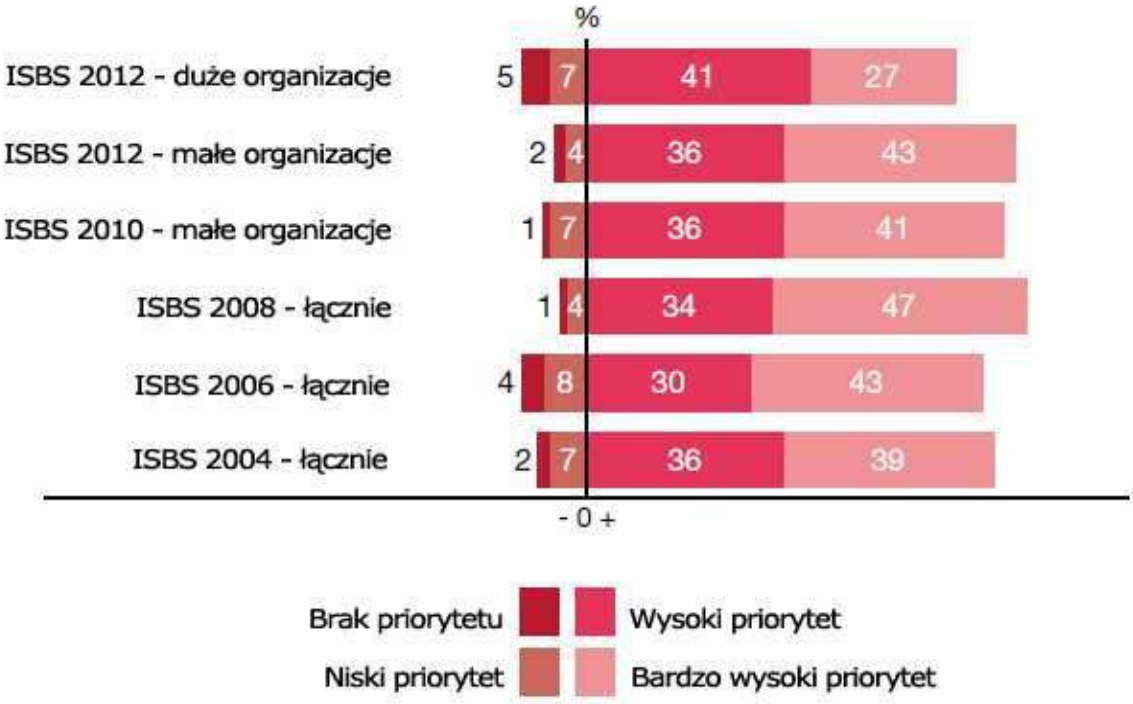
Organizacje wydają znaczące kwoty na bezpieczeństwo, gdy spodziewają się ataku. 8% budżetu IT jest średnio przydzielane na bezpieczeństwo informacji. 50% dużych firm spodziewa się wzrostu nakładu kosztów w następnym roku. 14% dużych firm spodziewa się spadku. 67% dużych firm oczekuje większej ilości prób włamań w następnym roku, natomiast 14% dużych organizacji oczekuję mniejszej ilości włamań.

W dużych organizacjach 12% respondentów twierdzi, iż kadra zarządzająca nadaje niski priorytet sprawom bezpieczeństwa. 20% twierdzi, że wydają mniej niż 1% budżetu IT na bezpieczeństwo informacji. Główną przyczyną takiego stanu rzeczy jest

to, że korzyści płynące z wydawania pieniędzy na bezpieczeństwo informacji są ciężko mierzalne. 80% dużych organizacji nie ocenia zwrotów z inwestycji w ich bezpieczeństwo komputerowe. 58% MŚP w ogóle nie próbuje oceniać efektywności ich wydatków na ochronę.

Strategie i audyt bezpieczeństwa

Wsparcie grupy zarządzającej jest niezbędne do efektywnego zarządzania bezpieczeństwem informacji w firmie. Na poniższym rysunku znajduje się wykres pokazujący jak zmieniał się priorytet bezpieczeństwa informacji dla menadżerów oraz kadry zarządzającej.



Rysunek 6. Zmiana priorytetu bezpieczeństwa wśród managerów w ciągu 8 lat. Źródło: opracowanie własne.

Trzy czwarte respondentów wierzy, że bezpieczeństwo jest ważne lub bardzo ważne dla ich szefostwa. Ten wynik jest bardzo zbliżony do wyniku sprzed dwóch lat. Dziewięciu na dziesięciu dyrektorów wykonawczych uważa, że nadają bezpieczeństwu IT wysoki priorytet. Jeden na ośmiu ludzi z personelu IT i bezpieczeństwa informacji czuje, że priorytet bezpieczeństwa w firmie jest niski. Często zarząd uważa, iż priorytet, który nadaję bezpieczeństwu gubi się w dużej organizacji.

Tak jak w przeszłości istnieje znaczne zróżnicowanie priorytetu bezpieczeństwa w firmach w zakresie działalności firmy. Firmy z branży usług finansowych, administracji, użyteczności publicznej i sektora technologii nadają

wysoki priorytet bezpieczeństwu. Jednakże największy priorytet bezpieczeństwa jest nadawany przez firmy, które działają w sektorze handlu detalicznego i dystrybucji, jest on dwa razy wyższy niż dla firm działających w sektorze nieruchomości i firm budowlanych. Nawiązując do trendu, który był dwa lata temu, małe i średnie przedsiębiorstwa są bardziej skłonne nadać wysoki priorytet bezpieczeństwu. Niektórzy respondenci w dużych organizacjach potępiają brak wysokiego priorytetu bezpieczeństwa i wpływ, który on posiada na działanie firmy. Na poniższym wykresie przedstawione są główne czynniki, którymi kierują się firmy by zabezpieczyć swoje dane.

Kultura bezpieczeństwa w firmie

Określenie celu organizacji w zakresie bezpieczeństwa ma zasadnicze znaczenie. Pracownicy personelu, którzy wiedzą, jakie jest ryzyko zwracają uwagę na sposób składowania danych oraz na działanie w przypadku pojawienia się przypadku naruszenia bezpieczeństwa danych. Ostania dekada pokazuje stały trend wzrostu wykorzystywania pisemnych zasad bezpieczeństwa. Ponad dwie trzecie małych i średnich przedsiębiorstw rozumie ten problem i posiada formalną kulturę bezpieczeństwa w firmie. Natomiast, prawie wszystkie duże organizacje mają ustaloną politykę bezpieczeństwa. Jedna na siedem firm, która nadała wysoki lub bardzo wysoki priorytet bezpieczeństwu, nie zapisała ich polityki bezpieczeństwa. Większość z nich to małe firmy, które polegają na słownym przekazie instrukcji i wierzą, że ich personel w pełni rozumie zasady polityki bezpieczeństwa.

Posiadanie zapisanej kultury bezpieczeństwa nie zapobiega atakom. Personel musi ją rozumieć i wprowadzić ją w życie. Tylko 26% respondentów z wprowadzoną polityką prywatności wierzy, że ich personel bardzo dobrze pojmuję kulturę bezpieczeństwa firmy. 21% sądzi, iż zrozumienie zasad bezpieczeństwa w ich firmie jest słabe.

Rysunek 7. Wykres przedstawiający główne czynniki zabezpieczania informacji poprzez firmy. Źródło: opracowanie własne.

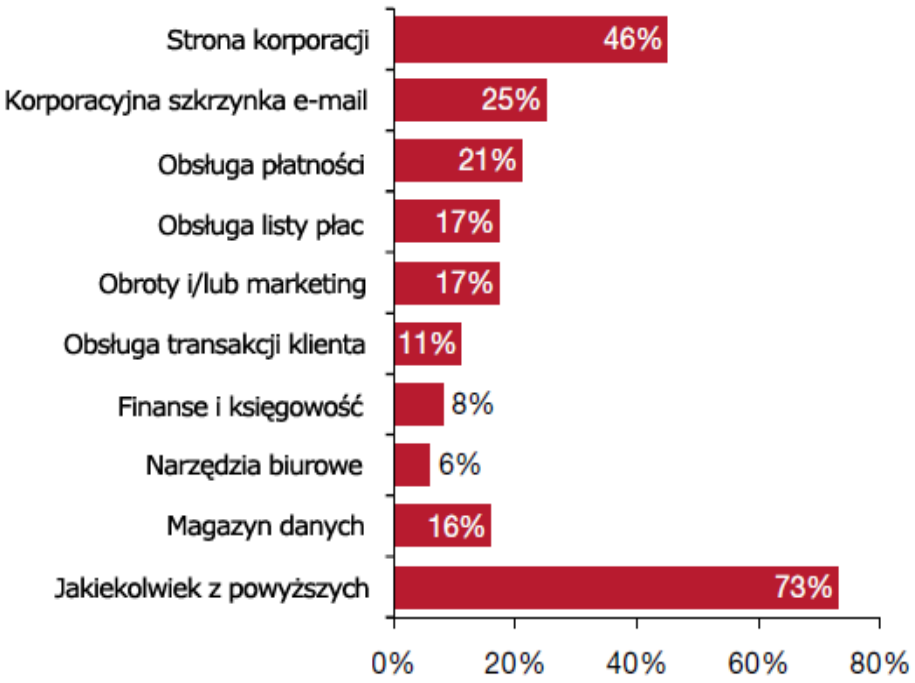
Na podstawie danych z „Information security breaches survey 2012”

Jaki jest główny czynnik określający wydatki na bezpieczeństwo IT?	
Ochrona informacji klienta	34%
Zapobieganie przestojom i przerwom	17%
Zgodność z prawami i rozporządzeniami	10%
Ochrona reputacji organizacji	12%
Utrzymanie integralności danych	8%
Ciągłość działalności firmy w przypadku awarii	4%
Ochrona własności intelektualnej	8%
Korzystanie z szans biznesowych	3%
Poprawa efektywności/redukcja kosztów	3%
Ochrona innych aktywów(np. gotówka) przed kradzieżą	1%
Razem	100%

Tabela 2. Wyniki ankiety dotyczącej głównych czynników określających wydatki na bezpieczeństwo. Źródło: opracowanie własne.

Cztery czynniki, które uzyskały największy wynik są identyczne do tych z 2010 roku. Najważniejszym czynnikiem jest ochrona informacji klienta. Zgodność z prawem i przepisami jest szczególnie istotna dla firm, które pracują w sektorze administracji oraz finansach. Respondenci, którzy skupiają się na efektywności i redukcji kosztów najczęściej odpowiadali, że poziom zabezpieczeń w firmie ma niski priorytet.

Zdalnie hostowane usługi mogą zaoszczędzić wiele pieniędzy, które są wydawane na serwery, licencje oraz konserwację systemu, szczególnie w małych i średnich przedsiębiorstwach. W dobie ograniczania kosztów można się spodziewać wzrostu liczebności tych usług. Około trzy czwarte respondentów używa, co najmniej jednej z usług, które są podane na wykresie poniżej. Poniższy wykres przedstawia, jakie usługi internetowe są outsourcowane do zewnętrznych dostawców.

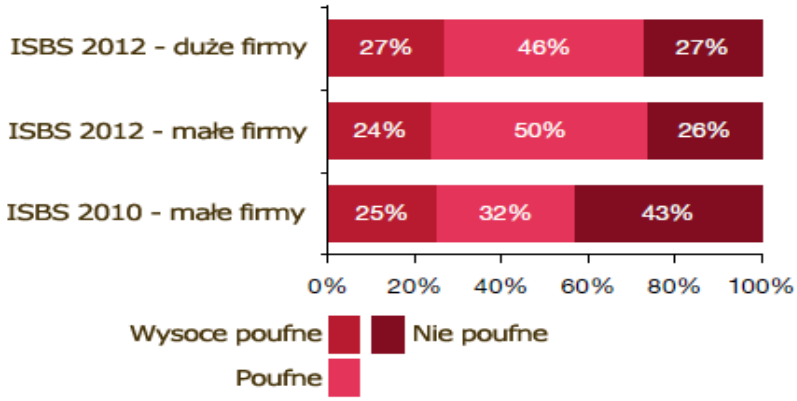


Rysunek 8. Usługi internetowe outsourcowane do zewnętrznych dostawców. Źródło: opracowanie własne.

Strony internetowe, skrzynki poczty elektronicznej, serwisy płatności są najczęściej używanymi usługami szczególnie w sektorze MŚP, gdzie ponad połowa stron i dwie piąte skrzynek e-mail jest hostowane na zewnętrznym serwerze. Natomiast w przypadku dużych firm, tylko 14% z nich używa zewnętrznych skrzynek

e-mail. Mniejsze firmy również częściej są skłonne używać oprogramowania biurowego i księgowego w wersji online.

Tajność danych umieszczonych w Internecie nie zmieniła się dużo w porównaniu z rokiem 2010. Około 25% dużych organizacji i około 20% małych firm, umieszcza na zewnętrznych serwerach ekstremalnie poufne informacje. Ponad 80% firm z branż produkcji, rozrywki, finansów i handlu detalicznego posiada tajne dane składowane w Internecie. Poniższy wykres przedstawia poufność danych umieszczanych w Internecie przez respondentów.



Rysunek 9. Poufność danych umieszczanych w Internecie przez respondentów. Źródło: opracowanie własne.

Jak widać na wykresie większość danych umieszczanych przez zarówno małe jak i duże firmy jest poufna. Ponad 80% firm działających w sektorze produkcji, rozrywki, sprzedaży detalicznej oraz organizacje zajmujące się finansami składuje poufne dane w Internecie.

Typy ataków komputerowych

Atak na system komputerowy to zamierzone działanie, które wykorzystuje lukę w zabezpieczeniach w celu uzyskania dostępu do informacji lub do zatrzymania działania usługi, która jest atakowana.

Głównym podziałem ataków na systemy informatyczne są ataki pośrednie oraz ataki bezpośrednie. Pierwszy typ dotyczy ataków gdzie osoba próbująca uzyskać nieautoryzowany dostęp do informacji używa rozproszonych systemów by przeprowadzić atak DDoS. Natomiast atak bezpośredni następuje wtedy, gdy osoba atakująca używa własnego systemu by zaatakować ofiarę.

Ataki komputerowe mogą być przeprowadzane na różnych płaszczyznach. Może być to atak na serwis komputerowy, podsłuchanie transmisji danych, wyłudzenie danych poprzez podszywanie się czy też na przykład użycie konia trojańskiego by uzyskać dostęp do mocy obliczeniowej komputera atakowanego. Pomysł i sukces ataku tak naprawdę zależy w dużej mierze od wiedzy oraz kreatywności osoby, która go przeprowadza.

Istota ataków komputerowych

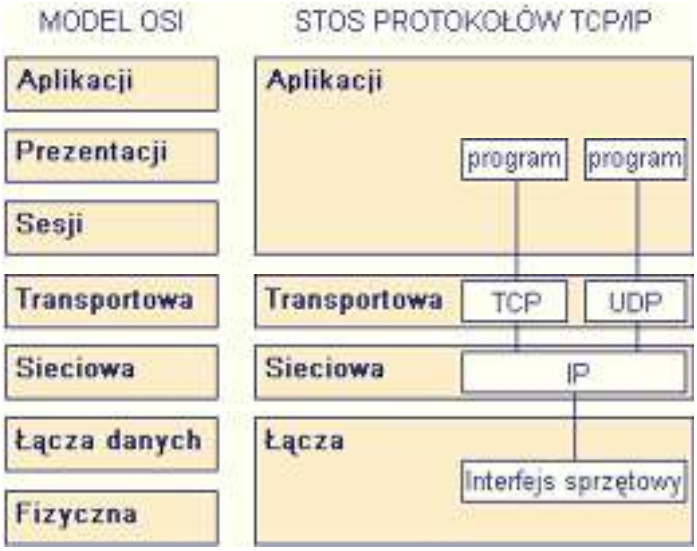
Aby zrozumieć istotę ataku komputerowego należy przyjrzeć się, w jaki sposób są składowane oraz przesyłane informacje pomiędzy aplikacjami.

Protokoły komunikacyjne

W Internecie przesyłanie informacji odbywa się za pomocą pakietów. Do komunikacji pomiędzy komputerami wymagany jest protokół. Dwa najbardziej znane i najczęściej używane protokoły komunikacyjne to TCP/IP oraz UDP.

Struktura protokołów TCP/IP

Model TCP/IP (ang. Transmission Control Protocol/Internet Protocol) – teoretyczny model warstwowej struktury protokołów komunikacyjnych. Model TCP/IP został stworzony w latach 70. XX wieku w DARPA, aby pomóc w tworzeniu odpornych na atak sieci komputerowych. Potem stał się on podstawą struktury Internetu.[8] Model stosu protokołów TCP/IP jest oparty na OSI.[9] Poniższy rysunek przedstawia warstwy modelu TCP/IP.



Rysunek 10. Model OSI oraz TCP/IP. Źródło: http://szmarcin.w.interia.pl/gif/tcpip_osi.gif (data odczytu 03.10.2012).

Atakujący może przeprowadzić atak na różnych poziomach modelu.

W warstwie łącza danych, osoba atakująca może wykorzystać ataki na tablicę MAC(MAC flooding), który powoduje przepełnienie pamięci przełącznika(ang. Switch). Po przepełnieniu bufora przełącznika, atakujący może użyć oprogramowania analizującego pakiety do wyodrębnienia interesujących dla niego danych. Kolejnym atakiem, który jest możliwy do przeprowadzenia w tej warstwie jest atak ARP Spoofing. ARP spoofing to atak sieciowy w sieci Ethernet pozwalający atakującemu

[8] Źródło: http://pl.wikipedia.org/wiki/Model_TCP/IP (data odczytu: 2.10.2012)

[9] OSI (ang. Open System Interconnection) lub Model OSI (pełna nazwa ISO OSI RM, ang. ISO OSI Reference Model – model odniesienia łączenia systemów otwartych) – standard zdefiniowany przez ISO oraz ITU-T opisujący strukturę komunikacji sieciowej.
Źródło: http://pl.wikipedia.org/wiki/Model_OSI (data odczytu: 2.10.2012)

przechwytywać dane przesyłane w obrębie segmentu sieci lokalnej. Przeprowadzony tą metodą atak polega na rozsyłaniu w sieci LAN odpowiednio spreparowanych pakietów ARP zawierających fałszywe adresy MAC. W efekcie pakiety danych wysyłane przez inne komputery w sieci zamiast do adresata trafiają do osoby atakującej pozwalając jej na podsłuchiwanie komunikacji. [10] Kolejnym typem ataku na tą warstwę jest atak DHCP spoofing. Atak ten polega na mieszaniu w komunikatach DHCP w celu uzyskania danych do konfiguracji sieci takich jak adres IP, maska, adres serwera DNS czy też adres bramy domyślnej.

W warstwie sieci wykorzystywane są jedne z najbardziej popularnych ataków, których pisałem w rozdziale „Typy Ataków Komputerowych" są to ataki DOS oraz DDoS.

Warstwa transportowa jest podatna na ataki skanowania portów. Atakujący może użyć skanera portów(na przykład nMap) by określić, jakie usługi i aplikacje są dostępne na maszynie osoby atakowanej. Do tego można określić, w jakim stanie znajduje się każdy port.

W ostatniej warstwie najbliższej użytkownikowi to jest warstwie aplikacji, obsługiwane są najważniejsze protokoły:

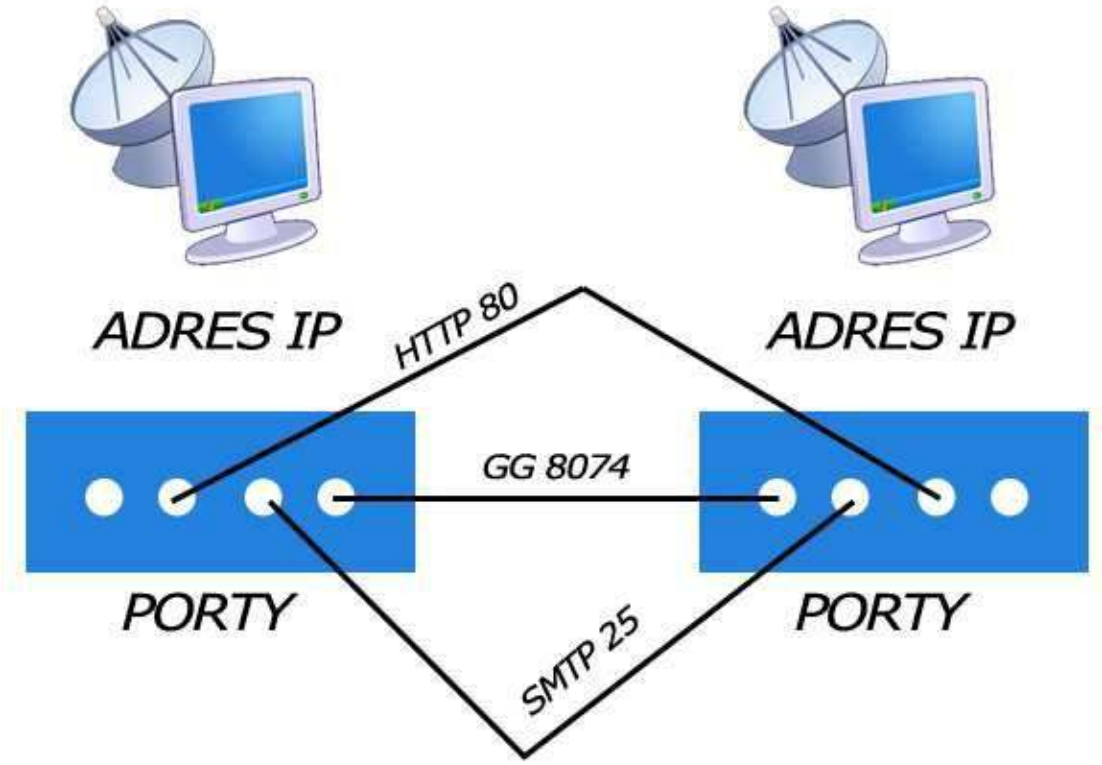
- HTTP – używany do przeglądania stron www
- FTP – obsługujący transfer plików przez Internet
- SMTP – służący do wysyłania elektronicznej korespondencji.

I wiele innych protokołów komunikacyjnych, z których korzystamy każdego dnia takich jak: skype, https, tftp, ssh, gg itd.

[10] Źródło: http://pl.wikipedia.org/wiki/ARP_Spoofing (data odczytu: 2.10.2012)

Gniazda komunikacyjne

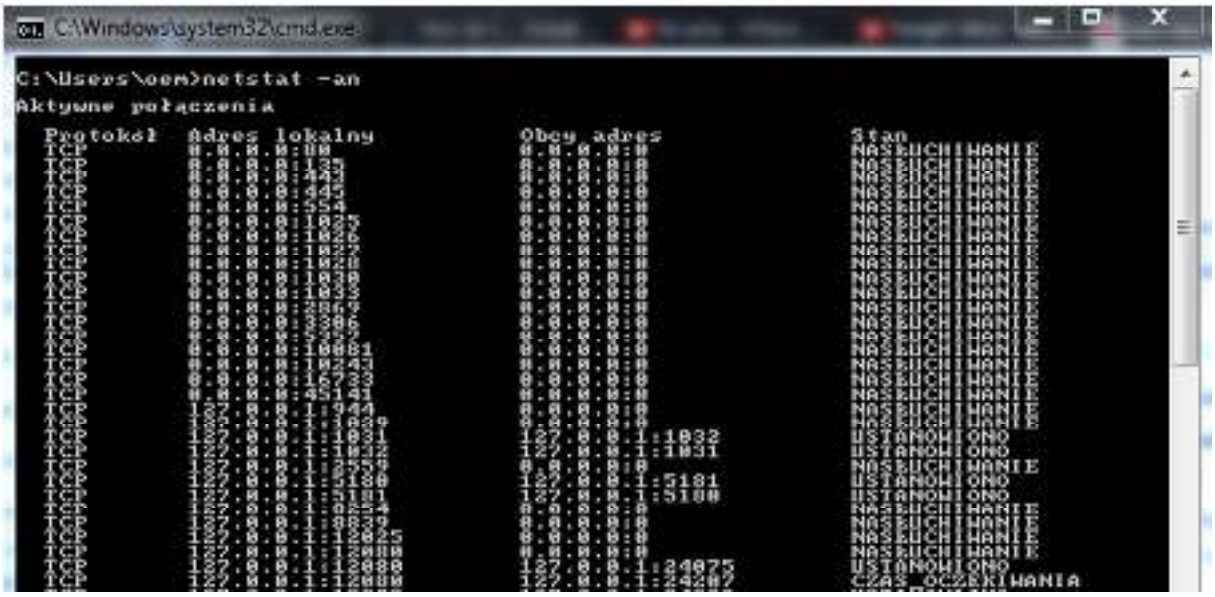
Gniazda komunikacyjne używane są w wewnętrznej wymianie informacji pomiędzy aplikacjami poprzez sieć komputerową. W dzisiejszych czasach większość komunikacji internetowej jest oparta na protokole IP, dlatego też większość gniazd sieciowych to gniazda wykorzystujące ten protokół. Poniższy rysunek prezentuje sposób łączenia się poprzez gniazda internetowe.



Rysunek 11. Sposób łączenia się poprzez gniazda sieciowe. Źródło: opracowanie własne.

Aby połączyć się poprzez gniazdo sieciowe należy utworzyć serwer, który akceptuje połączenia przychodzące na danym porcie używając określonego protokołu na przykład TCP lub UDP. Serwer ma za zadanie przyjmować połączenia od klientów. Nasłuchuje on na określonym porcie i w momencie prośby klienta o połączenie akceptuje ją. Po przyjęciu połączenia wymiana informacji następuje na określonym porcie bądź portach. W przypadku znanych usług na przykład przeglądarki internetowej, odbywa się to najczęściej na porcie 80 przy protokole http oraz 443 w

przypadku protokołu https. Na poniższym rysunku przedstawiłem przykłady połączeń z popularnymi usługami oraz porty, na których te usługi najczęściej pracują. Aby sprawdzić, jakie aplikacje powiązane są z określonymi gniazdami należy w systemie Windows lub systemach opartych o UNIX, użyć polecenia netstat. Poniższy rysunek przedstawia zrzut z programu netstat pokazujący połączenia.



Rysunek 12. Działanie komendy netstat w systemie Windows 7. Źródło: Opracowanie własne.

Na rysunku możemy zobaczyć dużą liczbę połączeń używających protokołu TCP, połączenia te są lokalne. Zauważalne są również porty połączenia oraz stan gniazda sieciowego. Komunikator, który napisałem działa w oparciu o protokół TCP na porcie 3435.

Kryptografia komunikacji sieciowej

Z uwagi na narastające potrzeby zabezpieczenia komunikacji sieciowej i poufnych informacji przesyłanych drogą sieciową, istotnym aspektem, na którzy powinno się zwrócić uwagę jest kryptografia. Kryptografia jest nauką poświęconą szyfrowaniu, sama w sobie jest mało przydatna, nabiera ona sensu wtedy, gdy staje się częścią jakiegoś większego systemu. Kryptografia ma za zadanie uniemożliwić nieautoryzowany dostęp do informacji, a także umożliwić go dla osób, które powinny takowy dostęp mieć. Odpowiednio zaszyfrowane dane, połączenie, może w znacznym stopniu utrudnić lub uniemożliwić nieautoryzowanej osobie dostęp do informacji. W tym rozdziale postaram się omówić najpopularniejsze algorytmy kryptograficzne, sposoby negocjacji klucza oraz uwierzytelniania użytkownika.

W kryptografii istnieje pewna forma nazewnictwa osób, które biorą udział w czynności kryptograficznej:

- Alice – jest to osoba, która pragnie wysłać wiadomość bezpiecznym kanałem
- Bob – jest osobą, która otrzymuje wiadomość od Alice
- Eve – jest to osoba, która pragnie podsłuchać wiadomość. Imię Eve pochodzi z angielskiego słowa „eavesdropper”, które w przetłumaczeniu na język polski oznacza osobę podsłuchującą.

Poniższy rysunek przedstawia klasyczny obraz przesyłu danych pomiędzy dwoma osobami, gdy istnieje trzecia, która chce dowiedzieć się, co zawiera wiadomość przesyłana.

Rysunek 13. Przesyłanie informacji pomiędzy Alice, a Bobem z udziałem osoby podsłuchującej. Źródło: opracowanie własne.

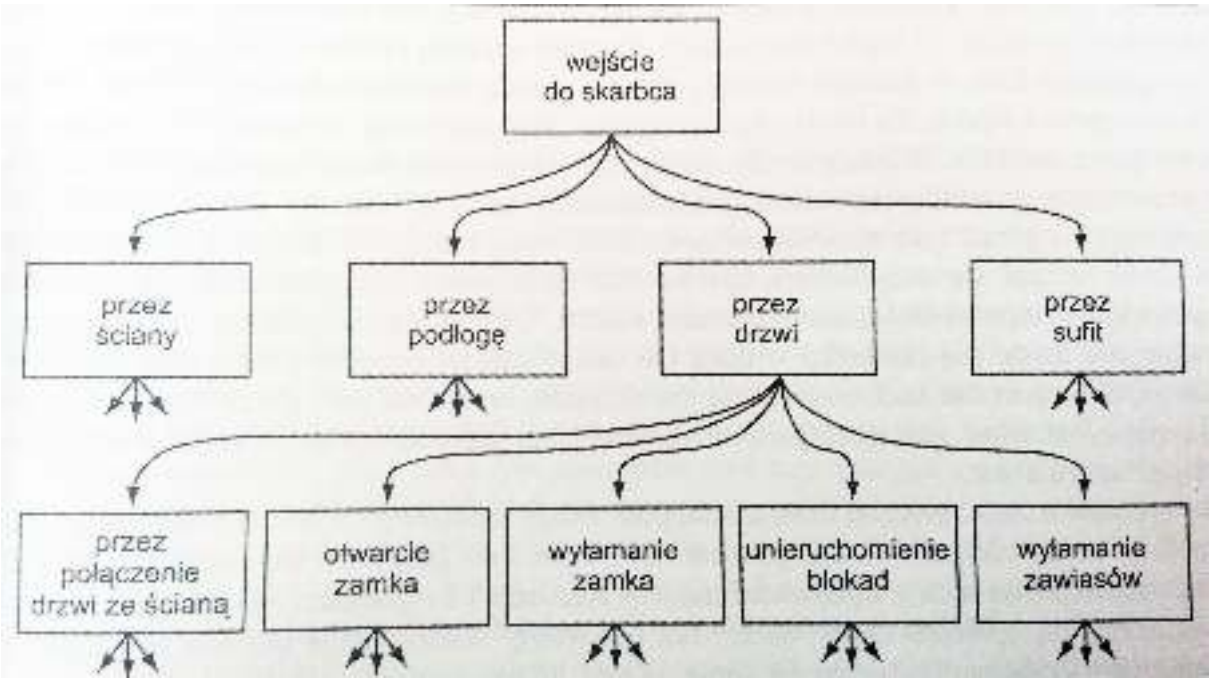
Narzuca się pytanie jak Alice powinna przesłać wiadomość do Boba, by podczas komunikacji osoba trzecia(Eve) niebyła skłonna ją przechwycić. Wiadomości wysyłane przez pomiędzy użytkownikami oznaczmy, jako *m*. aby Eve nie była w stanie dowiedzieć się, jaka wiadomość została przesłana pomiędzy Alice, a Bob powinniśmy ją zaszyfrować. W pierwszym etapie Alice i Bob muszą ustalić tajny klucz K_e. Klucz ten jest wykorzystywany do komunikacji kanałem, który może zostać podsłuchany przez Eve. Alice musi przesłać ten klucz do Boba na przykład używając poczty. Podczas przesyłania wiadomości *m* Alice musi ją najpierw zaszyfrować używając funkcji szyfrującej $E(K_e,m)$. Gdzie E – oznacza funkcje szyfrującą(ang. Encryption), K_e to klucz używany do szyfrowania, a *m* to wiadomość do zaszyfrowania. Aby odszyfrować wiadomość Bob musi użyć funkcji deszyfrującej $D(K_e,c)$. Gdzie D – oznacza funkcję deszyfrującą(ang. Decryption), K_e to klucz używany do deszyfrowania, natomiast *c* oznacza tekst zaszyfrowany(ang. Cipher). Po zastosowaniu funkcji deszyfrującej Bob otrzymuje tekst jawny wiadomości przesłanej przez Alice – *m.* Eve nie zna klucza K_e, dlatego gdy widzi ona zaszyfrowaną wiadomość nie jest w stanie jej odszyfrować.

Rysunek 14. Ogólna konfiguracja szyfrowania wiadomości pomiędzy Alice, a Bob-em. Źródło: opracowanie własne.

Dobre algorytmy kryptograficzne uniemożliwiają odszyfrowanie wiadomości bez znajomości klucza.

Prawo najsłabszego ogniwa

System zabezpieczeń nie jest mocniejszy niż jego najsłabszy element. System zabezpieczeń zwykle jest zbudowany z wielu elementów. Dobrze jest przyjąć założenie, iż atakujący znajdzie element systemu, który jest najbardziej podatny na atak. Sytuacja ta przypomina używanie łańcucha, w którym najsłabsze ogniwo pęka, jako pierwsze i nie ma znaczenia jak mocne są pozostałe ogniwa. Poprawienie systemu bezpieczeństwa wymaga wzmocnienia ogniwa, które jest najsłabsze, aby to zrobić należy wyodrębnić poszczególne składowe systemu i wskazać to najsłabsze. Autorzy książki Kryptografia w praktyce, sugerują by zestawić wszystkie ogniwa systemu w formie drzewa ataku.



Rysunek 15. Przykład drzewa ataku na skarbiec. Źródło: (zob. [1], str. 23).

Prawo najsłabszego ogniwa ma wpływ na pracę kryptografów. Na przykład wygodnie jest przyjąć założenie, iż użytkownicy wybierają hasła z rozsądkiem. W rzeczywistości tak nie jest. Przeciętny użytkownik wybiera hasło, które jest krótkie i łatwe do zapamiętania lub gdy system generuje hasła to wtedy je drukują i przyklejają na karteczce. W taki sposób można wyodrębnić słabe ogniwo, którym jest użytkownik nieświadomy swoich działań, które powodują znaczne osłabienie systemu bezpieczeństwa. Atakujący nie zawsze może wybrać miejsce ataku, które na pierwszy

rzut oka wydaje się najsłabsze. Z uwagi na różnych poziom doświadczeń i umiejętności atakującego, pewne rzeczy mogą wydawać mu się łatwiejsze. Dlatego powinno się stale ulepszać i poprawiać system zabezpieczeń.

Zasada Kerckhoffsa

Aby odszyfrować zaszyfrowany tekst Bob potrzebuje dwóch rzeczy. Pierwszą z nich jest algorytm szyfrujący *D*, a drugą klucz K_e. Istotne w kryptografii jest przestrzeganie zasady Kerckhoffsa: bezpieczeństwo systemu szyfrowania ma zależeć jedynie od ukrycia klucza K_e, a nie od ukrywania algorytmów. Można to uzasadnić w dość prosty sposób. Wymiana jednego algorytmu na inny może być czasami dość skomplikowanym procesem. Algorytmy najczęściej są wbudowane w sprzęt lub oprogramowanie, a gdy ktoś zdecyduje na jakiś algorytm to jest on używany przez długi czas. Otrzymanie algorytmu od osoby obsługującej system nie jest trudne. Eve może być też twórcą systemu kryptograficznego znającą algorytm szyfrowania. Dlatego właśnie bezpieczeństwo kryptograficzne powinno opierać się na tajemnicy klucza, a nie algorytmu.

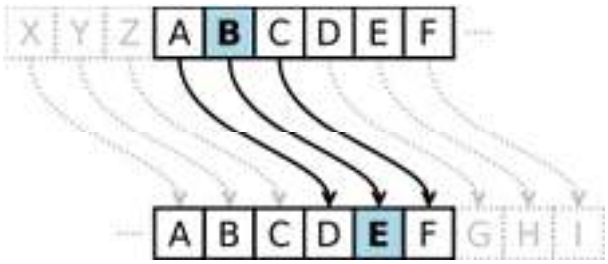
Algorytmy szyfrowania symetrycznego

Algorytmy szyfrowania symetrycznego używają do szyfrowania i deszyfrowania takiego samego klucza. Podstawową zaletą takich algorytmów jest szybkość ich działania.

Szyfr Cezara

Algorytmy szyfrowania symetrycznego były wykorzystywane przez rządzących w Starożytnym Rzymie. Pierwszym najprostszym algorytmem szyfrowania jest algorytm, który swoją nazwę wziął od Juliusza Cezara, który go wymyślił. Algorytm, o którym piszę to Szyfr Cezara.

Jest to przykład prostego algorytmu symetrycznego, który działa na zasadzie szyfru podstawieniowego. Każda litera w tekście jawnym jest zastępowana inną literą oddaloną w określonym kierunku o k znaków w alfabecie.



Rysunek 16. Schemat działania Szyfru Cezara. Źródło: http://pl.wikipedia.org/ (data odczytu 5.10.2012).

Szyfr ten w dzisiejszych czasach nie oferuje praktycznie żadnego bezpieczeństwa danych, gdyż jest on podatny na atak statystyczny. Polegający na analizie częstości występowania znaków w danej próbie.

Data Encryption Standard(DES)

DES jest to algorytm symetrycznego szyfrowania blokowego. Został zaprojektowany w 1975 roku przez firmę IBM na zlecenie Narodowego Biura Standardów USA(obecnie NIST[11]). DES używa szyfrowania 64 bitowymi blokami. Do szyfrowania i deszyfrowania wykorzystywane jest 56 bitów klucza. W ciągu 64 bitowym, co ósmy bit jest bitem kontrolnym.

Algorytm szyfrowania danych jest następujący: na początku tekst jawny, który ma zostać zaszyfrowany, dzielony jest na bloki 64-bitowe. Następnie dla każdego bloku wykonywane są następujące operacje:

1. Dokonywana jest permutacja początkowa bloku przestawiająca bity w pewien określony sposób – nie zwiększa ona bezpieczeństwa algorytmu, a jej początkowym celem było ułatwienie wprowadzania danych do maszyn szyfrujących używanych w czasach powstania szyfru.
2. Blok wejściowy rozdzielany jest na dwie 32-bitowe części: lewą oraz prawą.
3. Wykonywanych jest 16 cykli tych samych operacji, zwanych funkcjami Feistela, podczas których dane łączone są z kluczem. Operacje te wyglądają następująco:
    i. Bity klucza są przesuwane, a następnie wybieranych jest 48 z 56 bitów klucza.
    ii. Prawa część danych rozszerzana jest do 48-bitów za pomocą permutacji rozszerzonej.
    iii. Rozszerzona prawa połowa jest sumowana modulo 2 z wybranymi wcześniej (i przesuniętymi) 48 bitami klucza.

[11] National Institute of Standards and Technology (ang. Narodowy Instytut Standaryzacji i Technologii) to amerykańska agencja federalna spełniająca funkcję analogiczną do Głównego Urzędu Miar. W latach 1901 - 1988 nosił nazwę National Bureau of Standards (ang. Narodowe Biuro Standaryzacji). Główna siedziba znajduje się w Gaithersburg w stanie Maryland. Bezpośredni nadzór nad agencją sprawuje Departament Handlu Stanów Zjednoczonych (Department of Commerce). Źródło: http://pl.wikipedia.org/wiki/National_Institute_of_Standards_and_Technology (data odczytu: 8.10.2012)

iv. Zsumowane dane dzielone są na osiem 6-bitowych bloków i każdy blok podawany jest na wejście jednego z S-bloków (pierwszy 6-bitowy blok na wejście pierwszego S-bloku, drugi 6-bitowy blok na wejście drugiego S-bloku, itd.). Pierwszy i ostatni bit danych określa wiersz, a pozostałe bity kolumnę S-BOXa. Po wyznaczeniu miejsca w tabeli, odczytuje się wartość i zamienia na zapis dwójkowy. Wynikiem działania każdego S-bloku są 4 bity wyjściowe – tworzą one 32-bitowe wyjście S-bloków. Każdy S-Blok ma inną strukturę.
v. Wyjście S-bloków poddawane jest permutacji w P-blokach.
vi. Bity tak przekształconego bloku sumowane są z bitami lewej połowy danych.
vii. Tak zmieniony blok staje się nową prawą połową, poprzednia prawa połowa staje się natomiast lewą połową - cykl dobiega końca.

4. Po wykonaniu 16 cykli operacji lewa i prawa połowa danych jest łączona za pomocą operacji XOR.
5. Dokonywana jest permutacja końcowa.

Deszyfrowanie polega na zastosowaniu tych samych operacji w odwrotnej kolejności (różni się od szyfrowania tylko wyborem podkluczy, który teraz odbywa się od końca).[12]

[12] Źródło: http://pl.wikipedia.org/wiki/Data_Encryption_Standard (data odczytu: 8.10.2012)

TDEA

Potrójny DES, jest to modyfikacja algorytmu DES, która polega na trzykrotnym przetworzeniu tekstu jawnego algorytmem DES. Został opublikowany w 1998 roku. Jego nazwa wywodzi się z angielskiego skrótu Triple Data Encryption Algorithm i tak zapisana jest w normach ANSI X3.92, ISO/IEC 18033-3:2005, FIPS PUB 46-3. W notacji często używana jest forma 3DES lub TDES.

1. Na początku dane są szyfrowane pierwszym kluczem
2. Potem dane są deszyfrowane drugim kluczem.
3. W ostatnim kroku dane są szyfrowane trzecim kluczem.

3DES ma siłę 168 bitów poprzez trzykrotne szyfrowanie kluczem 56 bitowym. Realna siła algorytmu z uwagi na atak „Meet in the middle" wynosi 2^{112}.

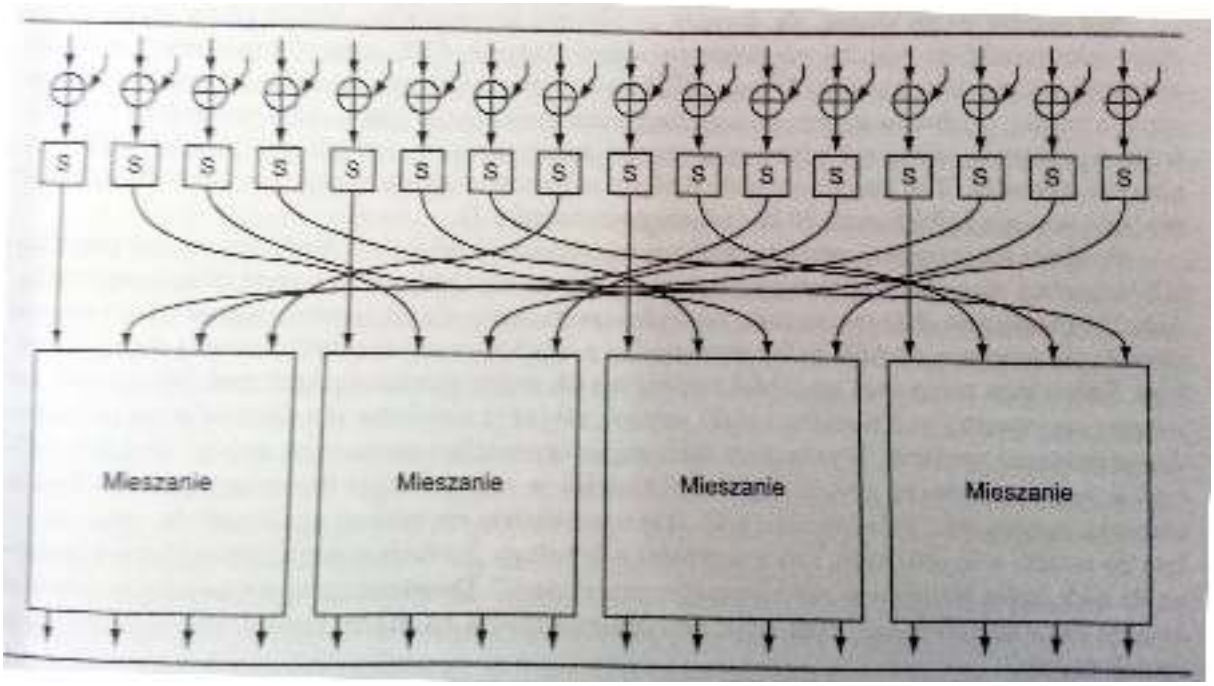
W potrójnym DES można użyć 3 trybów ustalania klucza.

1. Wszystkie klucze są od siebie niezależne.
2. Klucz 1 oraz klucz 2 są niezależne natomiast klucz trzeci jest taki sam jak klucz pierwszy.
3. Wszystkie klucze są identyczne.

Opcja pierwsza jest najsilniejsza. Trzy różne klucze pomnożone przez 56 bit każdy daje 168 niezależnych bitów. Opcja druga dostarcza mniejszego bezpieczeństwa. Dwa klucze pomnożone przez 56 bit dają 112 bitów służących do szyfrowania. Opcja ta jest silniejsza niż zwykły DES, ponieważ zapobiega atakowi „Meet in the middle". Opcja trzecia została stworzona po to by, 3DES był kompatybilny z DES. Zapewnia ona bezpieczeństwo na poziomie 56 bitów i tak naprawdę użycie tej opcji jest równoznaczne z użyciem algorytmu DES.

AES

Nazwa AES wywodzi się z angielskiego wyrażenia „Advanced Encryption Standard”. Czyli w tłumaczeniu na język polski: zaawansowany standard szyfrowania. AES to symetryczny szyfr blokowy, który został przyjęty przez NIST w wyniku konkursu ogłoszonego w 1997 roku. Jego początkowa nazwa to Rijndael. Twórcami tego algortymu są dwaj belgijscy kryptolodzy: Daemen i Vincent Rijmen. Podstawową różnicą pomiędzy AES, a DES jest to, iż ten pierwszy nie działa w oparciu o funkcje Feistela.



Rysunek 17. Pojedyncza tura szyfrowania algorytmem AES. Źródło: (zob. [1] str. 55).

Przy szyfrowaniu AES najpierw wykonywana jest operacja XOR z 128 bitami klucza tury. Bajty potem przestawiane są w pewnym porządku. Następnie za pomocą liniowej funkcji mieszającej, bajty są mieszane w grupach po cztery. W taki sposób działa pojedyncza tura AES. Całkowite szyfrowanie opiera się na 10-14 tur w zależności od długości klucza. Podobnie jak w DES, w AES istnieje pewien schemat do generowania kluczy jednak jest on innych w tych dwóch algorytmach. By zaatakować algorytm AES atak wymagałby kroków rzędu 2^{120} oraz 2^{100} bajtów pamięci. Taki zasób pamięci umożliwiłby zredukowanie poziomu bezpieczeństwa AES do 120 bitów. Takie

ataki nie są jeszcze dzisiaj możliwe i prawdopodobnie nie będą w ciągu najbliższych 50 lat. AES obsługuje klucze o długości 128, 192 oraz 256 bitów. AES jak powszechnie wiadomo jest standardem rządowym USA, nawet gdyby został złamany, pozwoli nam uniknąć wielu nieporozumień i problemów, ponieważ nikt nie powinien winić programisty za korzystanie ze standardowego algorytmu szyfrującego.

Tryby szyfrów blokowych

Algorytmy, które należą do grupy szyfrów blokowych potrafią szyfrować bloki, które są określonego rozmiaru. W praktyce zachodzi potrzeba by szyfrować dane o różnej długości. Aby to było możliwe należy użyć, któregoś z trybów szyfrowania blokowego.

Szyfr blokowy polega na zamianie tekstu otwartego *P*(ang. Plain) na tekst zaszyfrowany *C*(ang. Cipher), teksty te muszą mieć taką samą długość. Większość trybów wymaga by długość tekstu otwartego *P* była wielokrotnością długości bloku do szyfrowania. Aby uzyskać taki efekt tekst, który jest różny od rozmiaru bloku powinien zostać dopełniony. Istnieje wiele sposobów na dopełnianie tekstu, najprostszy z nich to dopisywanie zer na końcu wiadomości. Takie rozwiązanie nie jest dobre, ponieważ w pewnych sytuacjach, gdy wiadomość zawiera zera, to dopełnienie jest nieodwracalne. To znaczy, że po deszyfrowaniu wiadomości otrzymamy inną wiadomość niż ta, która została zaszyfrowana. Jeżeli przyjmiemy, że *P* jest tekstem otwartym, a l(*P*) to funkcja obliczająca długość tekstu w bajtach. To, aby uzyskać dobre dopełnienie należy dodać pojedynczy bajt o wartość 128, a potem tyle bajtów zerowych ile potrzebne jest do uzyskania długości bloku do szyfrowania. Po odszyfrowaniu tekstu zaszyfrowanego należy usunąć dopełnienie.

Przyjrzyjmy się teraz samym trybom szyfrowania blokowego. Najprostszym z nich jest tryb *elektronicznej książki kodów* (ECB ang. Electronic Code Book). Tryb ECB szyfruje każdy blok wiadomości osobno. Standardowy tryb ECB ma postać:

$$C_i = E(K, P_i) \text{ dla } i = 1, \ldots., k$$

Jego główną wadą jest to, iż gdy zaszyfrujemy pewną wiadomość *m* dwukrotnie to za każdym razem otrzymamy taki sam szyfrogram *c*.

Kolejnym trybem szyfrowania blokowego jest *tryb łańcuchowego szyfru blokowego*(CBC ang. Cipher Block Chaining), który jest najczęściej używanym trybem szyfrów blokowych. CBC różni się tym od ECB, że dzięki przekształcaniu każdego bloku

tekstu otwartego za pomocą funkcji XOR eliminuje on problemy występujące w trybie ECB. Użycie funkcji XOR na kolejnych blokach znacznie ogranicza ilość informacji, które są dostępne dla osoby atakującej. Działanie trybu CBC obrazuje poniższe równanie:

$$Ci = E(K, P_i \oplus C_{i-1}) \text{ dla } i = 1, \ldots., k$$

Jako C_0 używany jest wektor inicjujący IV(ang. initialization vector). Wartość wektora powinna być zmienna, gdyż użycie takiego samego wektora dla dwóch wiadomości dałoby taką samą wiadomość zaszyfrowaną.

Tryb OFB, jest to tak zwane *sprzężenie zwrotne*(ang. Output Feedback). Jest to metoda szyfrowania strumieniowego. OFB definiuję następujące równanie.

$$K_0 = IV$$

$$K_i = E(K, K_{i-1}) \text{ dla } i = 1, \ldots, k$$

$$C_i = P_i \oplus K_i$$

W tej metodzie szyfrowania również jest używany wektor inicjujący oznaczamy, jako K_0, który jest używany do generowania strumienia klucza K_1, …., K_k. Strumień klucza wraz z tekstem otwartym służą, jako argumenty funkcji XOR, która zwraca nam tekst zaszyfrowany.

Ostatnim trybem szyfrowania blokowego, który omówię jest tryb CTR. Czyli *tryb licznika*(ang. Counter Mode). Podobnie jak OFB jest to tryb szyfrowania strumieniowego, który jest zdefiniowany następującym równaniem:

$$K_0 = E(K, \text{Jednorazowy\_numer} \wedge i) \text{ dla } i = 1, \ldots, k$$
$$C_i = P_i \oplus K_i$$

Podobnie jak w przypadku innych szyfrów strumieniowych, na początku powinno nadać się niepowtarzalny numer jednorazowy, który jest najczęściej numerem kolejnej wiadomości uzupełnionym o dane zapewniające mu

niepowtarzalność. Strumień klucza generowany jest za pomocą łączenia numeru jednorazowego z licznikiem, a potem szyfrowaniu go w pojedynczy blok klucza. Aby wszystko działało licznik i numer jednorazowy muszą mieć długość bloku strumienia klucza. W celu zapewnienia dobrego poziomu bezpieczeństwa należy zadbać by kombinacja numer jednorazowy złączona z kluczem była niepowtarzalna. Tryb CTR jest bardzo prosty w użyciu, nie wymaga dopełniania wiadomości, jest prosty w implementacji. CTR ma jedną istotną wadę. W porównaniu z innymi algorytmami szyfrowania blokowego podczas sytuacji powtórzenia numeru jednorazowego, CTR pozwala na większy wyciek danych.

Kryptografia szyfrowania asymetrycznego

Kryptografia asymetryczna jest to rodzaj szyfrowania danych, które wykorzystuje architekturę klucza publicznego oraz prywatnego. Dane szyfrowane są kluczem publicznym, natomiast odszyfrowywanie następuje poprzez użycie klucza prywatnego. Aby zrozumieć działanie kryptografii asymetrycznej należy przyjrzeć się podstawom, na których jest ona oparta. Teoria kryptografii asymetrycznej opiera się na liczbach pierwszych. Liczba pierwsza jest liczbą naturalną, która nie posiada innych dzielników niż 1 oraz samą siebie. W matematyce istnieje kilka twierdzeń na temat liczb pierwszych. Euklides twierdził, iż istnieje nieskończenie wiele liczb pierwszych. Jest to przydatne znaczenie w kontekście łamania kluczy algorytmów asymetrycznych, ponieważ im większa liczba pierwsza użyta do klucza, tym trudniejsze jest złamanie go. Kolejnym bardzo istotnym twierdzeniem jest podstawowe twierdzenie arytmetyczne, które mówi: *Każdą liczbę całkowitą większą od 1 można przedstawić w dokładnie jeden sposób, jako iloczyn liczb pierwszych.* Dowód tego twierdzenia można znaleźć w praktycznie każdej książce traktującej o teorii liczb.

Do generowania małych liczb pierwszych używany jest algorytm o nazwie *Sito Eratostenesa*. Algorytm ten został stworzony ponad 2000 lat temu przez *Eratostenesa*, który był przyjacielem Archimedesa. Algorytm jest oparty na prostym pomyśle. Każda liczba złożona *c* jest podzielna przez liczbę pierwszą mniejszą od *c*. Małe liczby pierwsze są wykorzystywane do generowania dużych liczb pierwszych. W kryptografii głównymi operacjami wykonywanymi na liczbach pierwszych są operacje dzielenia, dodawania, odejmowania modulo. Zakładając, że *p* to liczba pierwsza, natomiast *r* to wynik. Aby obliczyć 25 modulo 7 należy najpierw podzielić 25 przez 7 co daje 3 z resztą 4, która jest odpowiedzią. Kolejnym istotnym faktem jest, iż każda liczba całkowita brana modulo *p* należy do zbioru 0, …, p-1, nawet, jeśli liczba pierwotna jest liczbą ujemną. Do operacji dzielenia modulo wykorzystywany jest *rozszerzony algorytm Euklidesa*, który wykorzystywany jest do obliczenia *największego wspólnego dzielnika(NWD)*.

RSA

Algorytm RSA jest najpowszechniej używanym algorytmem wykorzystującym architekturę klucza publicznego i prywatnego. Obsługuje on cyfrowe podpisy, certyfikaty, komunikację i jest on bardzo uniwersalnym narzędziem kryptograficznym. RSA opiera się na problemie faktoryzacji. RSA został stworzony przez Ronalda Rivesta, Adi Shamira i Leonarda Adlemana i od pierwszych liter ich nazwisk wziął on swoją nazwę. Do szyfrowania potrzebujemy dwóch kluczy. Klucza publicznego oraz prywatnego. W celu wygenerowania tych kluczy musimy zastosować algorytm.

1. Wybieramy losowo dwie duże liczby pierwsze p i q (najlepiej w taki sposób, aby obie miały zbliżoną długość w bitach, ale jednocześnie były od siebie odległe wartościami – istnieją lepsze mechanizmy faktoryzacji, jeżeli liczba ma dzielnik o wartości bliskiej $\sqrt{n}$)
2. Obliczamy wartość n = p*q
3. Obliczamy wartość funkcji Eulera dla n: $\varphi(n) = (p-1)(q-1)$
4. Wybieramy liczbę e (1 < e < ϕ(n)) względnie pierwszą z ϕ(n)
5. Znajdujemy liczbę d odwrotną do e mod ϕ(n): $d = e^{-1} \bmod \varphi(n)$

Klucz publiczny jest definiowany, jako para liczb (n, e), natomiast kluczem prywatnym jest para (n, d).

Aby odszyfrować i zaszyfrować wiadomość należy posłużyć się poniższymi wzorami.

Zanim zaszyfrujemy wiadomość, dzielimy ją na bloki o wartości liczbowej nie większej niż *n*, a następnie każdy z bloków szyfrujemy według wzoru:

$$c_i = m_i^e \mod n$$

Zaszyfrowana wiadomość będzie się składać z kolejnych bloków. Tak stworzony szyfrogram przekształcamy na tekst jawny, odszyfrowując kolejne blok według wzoru:

$$m_i = c_i^d \mod n$$

Możliwe jest także zaszyfrowanie wiadomości za pomocą klucza tajnego *d*, a następnie jej odszyfrowanie za pomocą klucza publicznego *e*. To właśnie ta własność sprawia, że RSA może zostać wykorzystany do cyfrowego podpisywania dokumentów.[13]

[13] Źródło: http://pl.wikipedia.org/wiki/RSA_(kryptografia) (data odczytu: 11.10.2012)

Protokół negocjacji klucza

Algorytm RSA jest algorytmem dostarczającym duże zasoby bezpieczeństwa, ale wolnym. Dlatego ważne jest by do komunikacji używać algorytmu symetrycznego. Problem, który powstaje przy takim rozwiązaniu to zagadnienie: jak przesłać klucz algorytmu symetrycznego do użytkownika, tak, aby osoba podsłuchująca nie poznała tego klucza. Do takich celów wykorzystuje się protokół negocjacji klucza. Poniższe rysunki przedstawiają bezpieczny sposób na negocjację klucza pomiędzy dwoma osobami.

Alice musi na początku wygenerować parę kluczy algorytmu RSA. Klucz publiczny zostaje przesłany otwartym tekstem Bobowi.

Rysunek 18. Przesyłanie klucza wygenerowanego klucza publicznego algorytmu RSA. Źródło: opracowanie własne.

Bob otrzymuje klucz publiczny należący do Alice i generuje klucz algorytmu szyfrowania symetrycznego. Następnie używa klucza publicznego Alice by zaszyfrować ten klucz i przesłać do Alice. Następnie zapisuje sobie ten klucz.

Rysunek 19. Przesyłanie zaszyfrowanego klucza algorytmu symetrycznego. Źródło: opracowanie własne.

Alice na podstawie swojego klucza prywatnego deszyfruje wiadomość od Boba i otrzymuje klucz algorytmu symetrycznego, który zapisuje.

Kolejne wiadomości wysłane pomiędzy Alice i Bobem są szyfrowane kluczem, który został ustalony pomiędzy Alice i Bobem.

Należy sobie zadać tutaj pytanie, a co z Eve. Otóż z uwagi na działanie algorytmu szyfrowania asymetrycznego Eve może poznać klucz publiczny Alice. Gdy Bob dostanie klucz Alice i zaszyfruje wiadomość i prześle ją z powrotem do Alice, Eve nie będzie w stanie odszyfrować wiadomości. Nawet gdyby Eve próbowała zmienić coś w kluczu, wtedy Alice nie będzie mogła odszyfrować wiadomości. Gdy Alice i Bob uzgodnią klucz, komunikacja będzie nadal szyfrowana tylko teraz za pomocą algorytmu symetrycznego, który jest o wiele szybszy w działaniu. Jest to kluczowe dla aplikacji, którą napisałem, ponieważ w niej, jako Alice występuje, jako serwer, Bob, jako klient, a Eve, jako potencjalny haker.

Konkurencyjne produkty wspierające szyfrowanie informacji

W Internecie istnieje wiele ciekawych programów, które wspierają bezpieczeństwo informacji. Możemy tam znaleźć programy służące do szyfrowania komunikacji głosowej oraz tekstowej, poczty elektronicznej, transmisji plików oraz inne ciekawe narzędzia. Jednakże nigdy nie będziemy pewni czy „bezpieczny program", który użytkujemy nie zawiera złośliwego kodu, który zamiast pomagać tak naprawdę nam szkodzi. Przykładem może być tutaj jeden z najpopularniejszych komunikatorów głosowych firmy Microsoft, o którym będę pisał w dalszej części tego rozdziału. Jeden z użytkowników programu Skype, który próbował uruchomić 64 bitową wersję komunikatora odkrył ciekawy błąd. Z uwagi na to, iż system Microsoft Windows w wersji 64 bit nie posiada programu NTVDM(ang. NT Virtual DOS Machine), który służy do uruchamiania starszych aplikacji pracujących jeszcze pod system Microsoft DOS, pojawił się błąd:

„The program or feature "C:\Documents and Settings\Myria\Local Settings\Temp\12\1.com" cannot start or run due to incompatibility with 64-bit versions of Windows. Please contact the software vendor to ask if a 64-bit Windows compatible version is available."

Okazało się, iż program Skype próbuje przy pomocy sprytnie ukrytego programu, wydobyć informacje na temat BIOS-u. Nie wiadomo, do czego te informacje mogą zostać wykorzystane i czy są gromadzone przez firmę rozwijającą komunikator.[14] Jednakże to pokazuje, iż nawet w przypadku bardzo solidnej firmy, jaką jest Microsoft, czasami może dochodzić do niewyjaśnionych wycieków informacji. W przypadku własnego programu lub programu w wersji Open Source jesteśmy w 100% pewni, z czym mamy do czynienia.

[14] Źródło: http://www.pcworld.pl/news/106372/Skype.zaglada.do.BIOSu.html (data odczytu: 13.03.2012)

Przykłady komunikatorów używających szyfrowania.

Z uwagi na to, że moja praca opiera się na aspekcie komunikacji postaram się przybliżyć konkurencyjne oprogramowanie, który może zostać użyte do podobnych celów jak program, który napiszę. W sieci dostępne jest oprogramowanie, które służy do szyfrowania komunikacji. Ze względu na treść przesyłanych informacji komunikatory dzielą się na następujące kategorie:

- komunikatory tekstowe są to komunikatory, które używają tekstu do wymiany informacji,
- multikomunikatory potrafią używać wielu protokołów komunikacyjnych do przesyłu informacji. Są to informacje zarówno tekstowe jak i na przykład głosowe lub wideo,
- komunikatory głosowe używają technologii przesyłania dźwięku do komunikacji,
- komunikatory wideo bazują na transmisji wideo podczas prowadzenia komunikacji.

Wiele komunikatorów łączy cechy wyżej opisanych kategorii komunikatorów w jedną całość.

Do najpopularniejszych programów używających kryptografii należą AQQ, Skype, Gadu Gadu oraz Pidgin.

- Pidgin należy do grupy multikomunikatorów[15]. Jest to program, który umożliwia komunikacje w kilku sieciach komunikatorów między innymi: AIM, ICQ, Google Talk, Jabber, MSN, Yahoo, Bonjour, IRC. Pidgin wspiera wiele funkcji, które oferują wspomniane wyżej protokoły komunikacyjne

[15] Multikomunikator internetowy to rodzaj komunikatora internetowego zawierającego w sobie metody komunikacji z użytkownikami wielu różnych komunikatorów. Gdy typowy komunikator umożliwia komunikację tylko między użytkownikami danego komunikatora tak multikomunikatory zapewniają komunikację z użytkownikami wielu komunikatorów. Przykładami mogą być AQQ, Digsby, Konnekt, Miranda IM, Pidgin (dawniej GAIM), Trillian, WTW, Nimbuzz, fring - dwa ostatnie przeznaczone przede wszystkim na platformy mobilne. Źródło: http://pl.wikipedia.org/wiki/Multikomunikator_internetowy (data odczytu 7.03.2012)

takie jak: przesyłanie plików, emotikony. Możliwe jest też dodanie nowych funkcjonalności poprzez pluginy. Pidgin jest darmowym komunikatorem, kod programu jest udostępniony na licencji GNU,

- AQQ jest multikomunikatorem polskiej produkcji. Oprócz standardowych funkcji takich jak pisanie tekstu, ustawianie statusów oraz opisów, umożliwia on przesyłanie plików oraz obrazów, konferencje. Możliwe jest również prowadzenie rozmów poprzez technologię VOIP[16]. Aqq wspiera protokoły: Jabber, Gadu-Gadu, Tlen, Facebook, NkTalk. Możliwe jest również rozszerzenie funkcjonalności programu poprzez wtyczki,
- Skype jest jednym z najbardziej popularnych komunikatorów umożliwiającym rozmowę głosową oraz wideo. Komunikator ten umożliwia za dodatkową opłatą rozmowy do sieci telefonii stacjonarnej oraz komórkowej. Do szyfrowania komunikacji Skype używa algorytmu AES 256 bit. Skype jest oparty na komunikacji P2P. Komunikator wspiera technologię VOIP. Oferuje również bezpośrednią wymianę plików. Skype ma ponad 200 milionów użytkowników na świecie oraz około 4 milionów w Polsce.

[16] VoIP (ang. Voice over Internet Protocol) – technologia cyfrowa umożliwiająca przesyłanie dźwięków mowy za pomocą łączy internetowych lub dedykowanych sieci wykorzystujących protokół IP, popularnie nazywana "telefonią internetową". Dane przesyłane są przy użyciu protokołu IP, co pozwala wykluczyć niepotrzebne "połączenie ciągłe" i np. wymianę informacji, gdy rozmówcy milczą. Źródło: http://pl.wikipedia.org/wiki/Voice_over_Internet_Protocol (data odczytu 7.03.2012)

Aplikacja komunikatora szyfrowanego

Jako przykład aplikacji, która wykorzystuje bezpieczne szyfrowanie, komunikację przedstawię napisany przeze mnie komunikator szyfrowany. Aplikacja wykorzystuje techniki i algorytmy, które przedstawiłem w różnych rozdziałach tej pracy. Jest to program używający wielowątkowości, programowania sieciowego oraz wielu zaawansowanych technik nowoczesnego programowania. Aplikacja została napisana w języku Java, o którym chciałbym napisać przed opisem aplikacji.

Java

Java jest to obiektowy język oprogramowania, który został stworzony przez grupę roboczą z firmy Sun Microsystems pod kierownictwem Jamesa Goslinga. Java jest językiem, który jest kompilowany do kodu pośredniego, który jest potem intepretowany przez maszynę wirtualną komputera, na którym program został uruchomiony. Takie podejście do kompilacji pozwala uruchamiać ten sam kod na wielu platformach i systemach operacyjnych bez potrzeby rekompilacji kodu. Najważniejsze cechy języka Java to:

- obiektowość,
- dziedziczenie,
- niezależność od architektury,
- sieciowość i obsługa programowania rozproszonego,
- niezawodność i bezpieczeństwo.

W Javie wszystkie obiekty są pochodną specjalnej klasy *Object*, z którego dziedziczą podstawowe zachowania i właściwości. Podstawowe rzeczy, które są dziedziczone to porównywanie obiektów, kopiowanie, niszczenie, identyfikacja oraz wsparcie dla programowania wielowątkowego. Java umożliwia dziedziczenie tylko z jednej klasy i z wielu interfejsów.

Z uwagi na to, iż kod Javy jest kompilowany do kodu pośredniego jest ona niezależna od architektury i systemu operacyjnego. Taki kod ma jednak swoje wady,

gdyż jest wolniejszy od kodu kompilowanego do postaci binarnej. Istnieją kompilatory, które kompilują kod Javy do postaci maszynowej, jednakże wtedy jest on zależny od platformy, na której został skompilowany.

Java jest w bardzo inteligentny sposób rozwiązane jest programowanie sieciowe. Język nie widzi różnicy pomiędzy danymi, które napływają ze strumienia pliku czy też z gniazda sieciowego. Java umożliwia pisanie apletów, które są umieszczane na stronach internetowych i uruchamiane przez przeglądarki internetowe.

Java jest zaprojektowana w taki sposób, by utrudnić programiście popełnienie błędów. Posiada ona system wyjątków, gdzie każdy wyjątek musi zostać „złapany” przez odpowiedni blok. Dzięki temu aplikacja unika nieoczekiwanego zamknięcia wskutek wyjątku aplikacji.

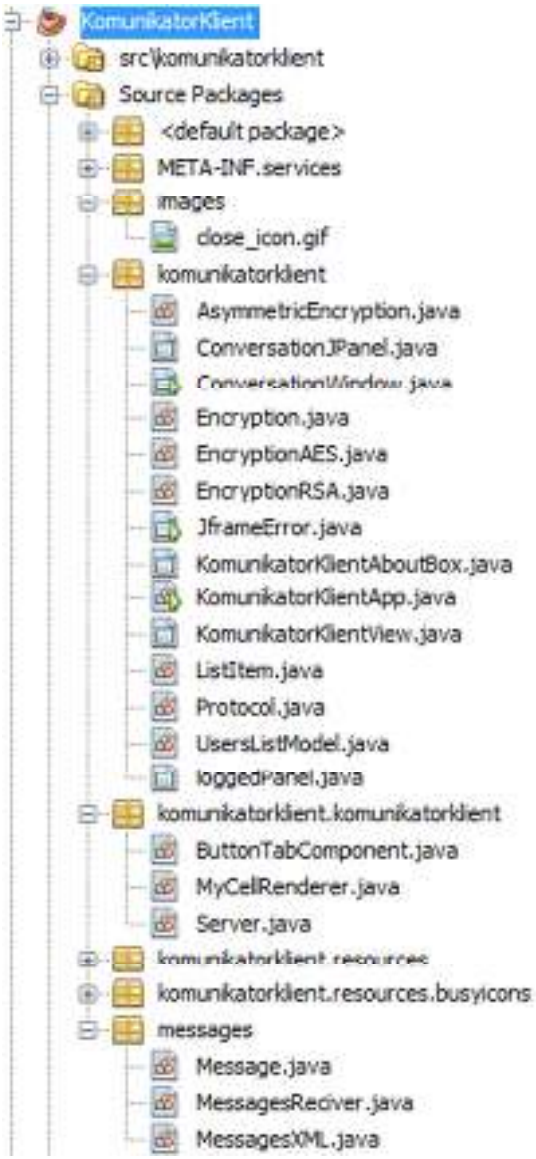
Struktura aplikacji

Aplikacja zbudowana jest z dwóch programów. Pierwszy program jest to klient, z którego korzysta użytkownik, natomiast drugą jest serwer uruchomiony w Internecie. Przedstawia to obrazek poniżej.

Rysunek 20. Struktura aplikacji klient-serwer. Źródło: http://upload.wikimedia.org/wikipedia/commons/thumb/f/fb/Server-based-network.svg/200px-Server-based-network.svg.png (data odczytu 12.10.2012).

Struktura oprogramowania klienckiego

Oprogramowanie klienckie zbudowane jest z klas, plików z obrazami, bibliotek oraz plików konfiguracyjnych. Na poniższym obrazie można zobaczyć jak ona wygląda.



Rysunek 21. Struktura programu klienckiego komunikatora szyfrowanego. Źródło: opracowanie własne.

Plik *AsymmetricEncryption.java* jest to interfejs, który dziedziczy z klasy *Encryption*. Interfejs wyznacza metody, które muszą zawierać klasy dziedziczące z niego by mogły być obsłużone przez metody używające algorytmów szyfrowania asymetrycznego.

Rysunek 22. Struktura interfejsu *AsymmetricEncryption*. Źródło: opracowanie własne.

Plik *ConversationJPanel.java* jest to element GUI[17], który jest odpowiedzialny za wyświetlenie panelu do rozmowy z użytkownikiem oraz obsługę wysyłania wiadomości i pokazywania, którzy użytkownicy są aktywni.

Plik ConversationWindow.java jest elementem GUI, który służy do przełączania zakładek rozmów użytkowników, wyświetla on elementy *ConversationJPanel* w zakładkach.

Interfejs zawarty w pliku Encryption.java służy do zamodelowania obiektów, które służą do szyfrowania danych. Z tego interfejsu korzystają klasy szyfrowania asymetrycznego oraz symetrycznego.

Rysunek 23. Struktura interfejsu *Encryption*. Źródło: opracowanie własne.

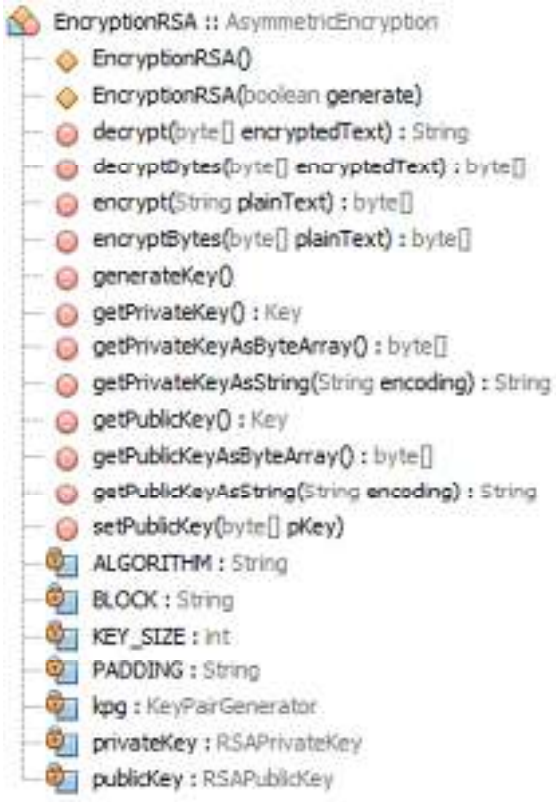
EncryptionAES.java jest to plik, który jest bardzo istotny z punkty widzenia bezpieczeństwa i kryptografii. To tutaj odbywa się szyfrowanie danych algorytmem AES. Klasa *EncryptionAES* dziedziczy z klasy *Encryption* zawiera metody, które służą do szyfrowania tekstu do ciągu bajtów, deszyfrowania ciągu bajtów na tekst, generowania klucza algorytmu AES oraz pobierania tego klucza. Klasa pozwala ustawić takie właściwości jak długość klucza, typ dopełniania algorytmu(w pracy jest

[17] Graficzny interfejs użytkownika (ang. Graphical User Interface, GUI), często nazywany też środowiskiem graficznym – ogólne określenie sposobu prezentacji informacji przez komputer oraz interakcji z użytkownikiem, polegające na rysowaniu i obsługiwaniu widżetów. Źródło: http://pl.wikipedia.org/wiki/Interfejs_graficzny (data odczytu: 12.10.2012)

to *PKCS5Padding*). Długość klucza ustawiona jest na 128 bitów z uwagi na to, że aby uruchomić AES z dłuższym kluczem należy instalować specjalnie rozszerzenie Java o nazwie Java Cryptography Extension(JCE), dla celów demonstracji i łatwości użytkowania aplikacji zdecydowałem się użyć klucza o długości 128.

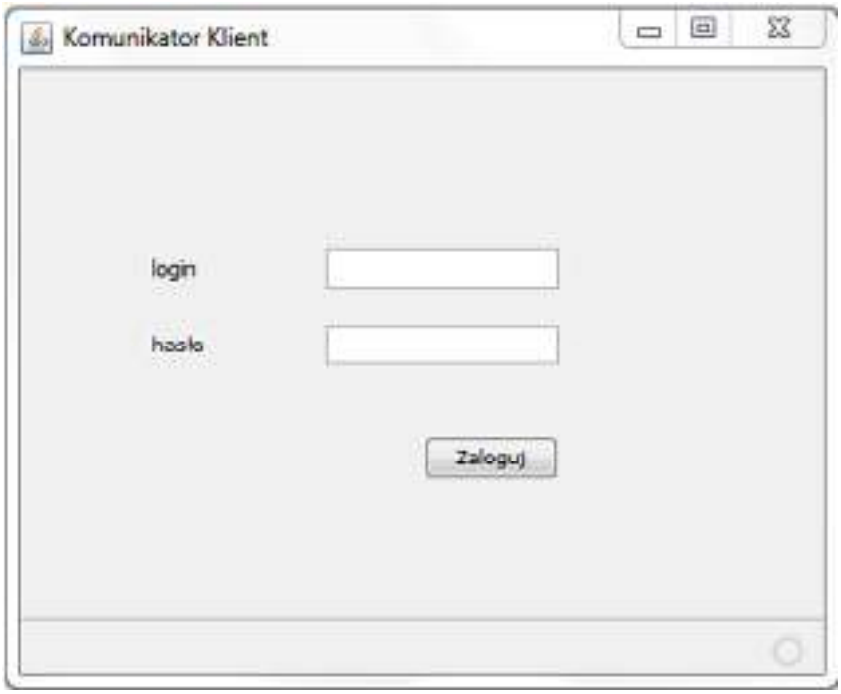
Rysunek 24. Struktura klasy *EncryptionAES*. Źródło: Opracowanie Własne.

Kolejnym plikiem bardzo istotnym z punktu widzenia bezpieczeństwa jest plik EncryptionRSA.java, w którym zawarte są metody odpowiedzialne za szyfrowanie, deszyfrowanie bajtów oraz stringów, generowanie kluczy publicznych i prywatnych, przekształcanie kluczy do formatu tekstowego oraz bajtowego. Na poniższym rysunku znajduje się struktura klasy EncryptionRSA.



Rysunek 25. Struktura klasy EncryptionRSA. Źródło: opracowanie własne.

Elementem programu, od którego aplikacja zaczyna się uruchamiać jest klasa *KomunikatorKlientApp*, uruchamia ona widok o nazwie *KomunikatorKlientView*, który jest odpowiedzialny za wyświetlenie pierwszego okna podczas startu programu z możliwością zalogowania.



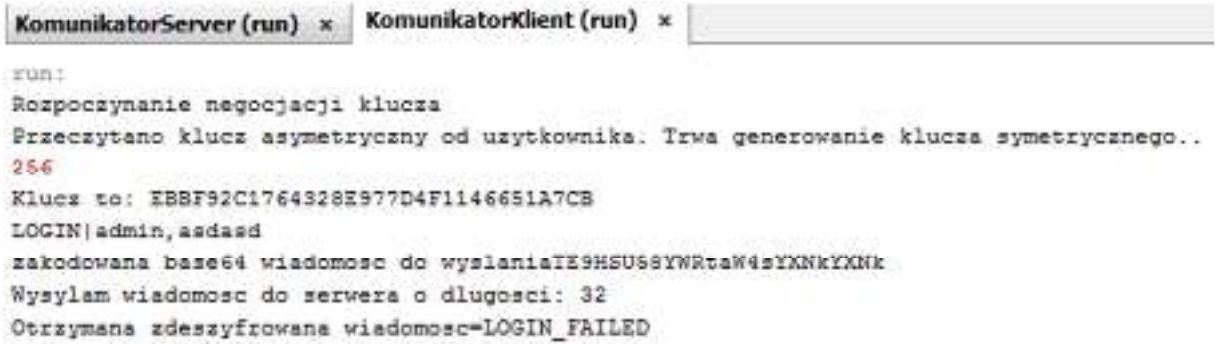
Rysunek 26. Panel służący do logowania się do komunikatora. Źródło: opracowanie własne.

Klasa ta łączy się również z serwerem za pomocą metody ConnectToServer, a po naciśnięciu przycisku logowania i wprowadzeniu danych rozpoczyna negocjacje klucza oraz wysyła po uzyskaniu klucza algorytmu symetrycznego, zaszyfrowany login i hasło użytkownika i w zależności od odpowiedzi serwera, albo loguje klienta, albo wyświetla komunikat o niepoprawnym logowaniu. Poniżej przedstawiam kod

programu, który realizuje połączenie z serwerem oraz negocjację klucza.

Rysunek 27. Fragment kodu programu obrazujący działanie negocjacji kluczy po stronie klienta. Źródło: opracowanie własne.

Powyższy kod wysyła komendy diagnostyczne, które przedstawiają, w jaki sposób odbywa się negocjacja klucza. Poniższy rysunek przedstawia zapis logów aplikacji klienckiej.



Rysunek 28. Zapis z wyjścia aplikacji klienckiej podczas niepoprawnego logowania do systemu. Źródło: opracowanie własne.

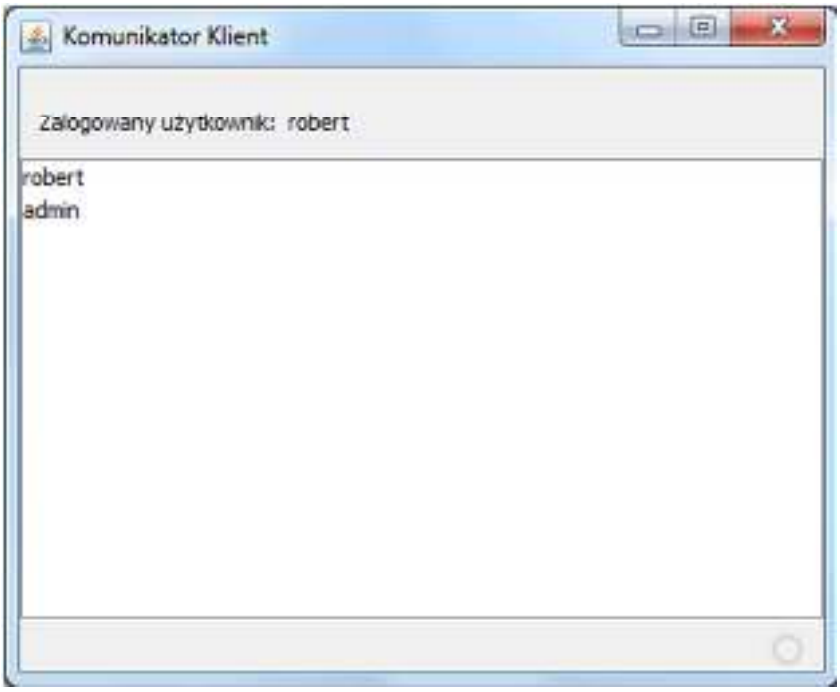
W przypadku poprawnej autoryzacji serwer wysyła do klienta wiadomość o poprawnym logowaniu *LOGIN_OK* oraz następnie wysyła mu w postaci XML[18] listę

[18] XML (ang. Extensible Markup Language, w wolnym tłumaczeniu Rozszerzalny Język Znaczników) – uniwersalny język formalny przeznaczony do reprezentowania różnych danych w strukturalizowany sposób. XML jest niezależny od platformy, co umożliwia łatwą wymianę dokumentów pomiędzy heterogenicznymi

użytkowników, którzy są online. Klient przetwarza tą listę i dodaje ją do okna użytkowników online.

Rysunek 29. Zapis z wyjścia aplikacji klienckiej podczas poprawnego logowania do systemu. Źródło: opracowanie własne.

Po poprawnej autoryzacji i otrzymaniu od serwera listy użytkowników w postaci XML, program kliencki wyświetla tą listę użytkowników.



Rysunek 30. Lista zalogowanych użytkowników w aplikacji klienckiej. Źródło: opracowanie własne.

(różnymi) systemami i znacząco przyczyniło się do popularności tego języka w dobie Internetu. XML jest standardem rekomendowanym oraz specyfikowanym przez organizację W3C. Źródło: http://pl.wikipedia.org/wiki/XML data odczytu: 10.11.2012

Za wyświetlanie użytkowników odpowiedzialna jest klasa *loggedPanel*. Jest to widok dla zalogowanego użytkownika. Poniżej przedstawię kod tej klasy wraz omówieniem ważnych elementów z punktu działania aplikacji.

```
30  public class loggedPanel extends javax.swing.JPanel
31  {
32      protected Server server;
33      protected EncryptionAES sEncryption;
34      protected Protocol protocol;
35      protected java.util.ArrayList<ListItem> usersOnline;
36      protected ConversationWindow conversation;
37      protected messages.MessagesReciver mr;
```

Rysunek 31. Definicja klasy *loggedPanel* wraz właściwościami tej klasy. Źródło: opracowanie własne.

Klasa *loggedPanel* jest klasa publiczną dziedziczącą z *javax.swing.JPanel*. Jej elementy to:

- *Server* server – jest to referencja do instancji serwera, która pozwala nam na uzyskanie ważnych funkcji do komunikacji.
- *EncryptionAES* sEncryption – jest to referencja do obiektu typu EncryptionAES, który umożliwia szyfrowanie symetryczne wiadomości.
- *java.util.ArrayList<ListItem>* usersOnline – jest to lista obiektów typu ListItem, przechowująca użytkowników, którzy są akutalnie dostępni
- *ConversationWindow* conversation – jest to referencja do okna rozmowy pomiędzy użytkownikami.
- *messages.MessagesReciver* mr – jest to referencja do klasy, która zajmuje się przetwarzaniem wiadomości oraz wysyłaniem ich do odpowiedniego okna. Klasa ta pracuje w osobnym wątku i zostanie opisana poniżej, gdyż z punkty działania aplikacji jest bardzo ważna do poprawnego jej działania.

```
40      public  loggedPanel()
41      {
42          initComponents();
43          server = Server.getInstance();
44          this.jLabel2.setText(server.getLoggedUserName());
45          sEncryption = server.getEncryption();
46
47          protocol = new Protocol();
48          protocol.setEncryption(sEncryption);
49          protocol.setStreams(server.inBuffered, server.outBuffered, server.in, server.out);
50
51          initList();
52          initListeners();
53
54          conversation  = new ConversationWindow();
            messages.MessagesReciver mr = new messages.MessagesReciver();
56
57          mr.setConversationWindow(conversation);
58          mr.setProtocol(protocol);
59          mr.setUsersOnline(usersOnline);
60          mr.setJList(jList1);
61          mr.setJScrollPane(jScrollPane1);
62          mr.execute();
63      }
```

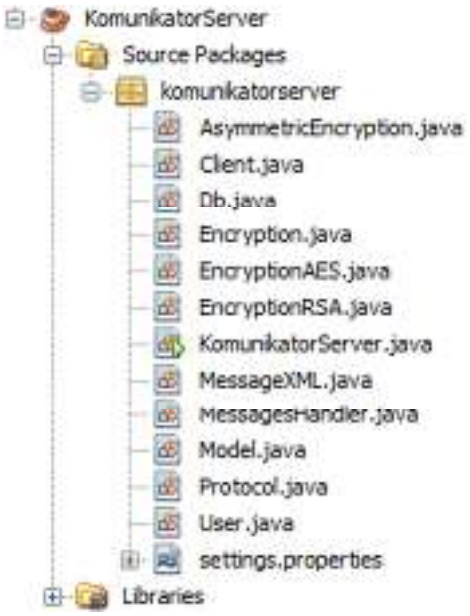
Rysunek 32. Konstruktor klasy *loggedPanel*. Źródło: opracowanie własne.

W konstruktorze klasy loggedPanel następuje zainicjowanie komponentów klasy dla GUI. Następnie jest popierana instancja serwera. Konstruktor zapisuje nazwę zalogowanego użytkownika w etykiecie na formie. Potem pobierana jest obiekt służący do szyfrowania komunikacji oraz protokół komunikacyjny. Po pobraniu tych elementów następuje zainicjowanie listy użytkowników online oraz nasłuchiwaczy do elementów formy. Tworzony jest obiekt konwersacji, który w przypadku nowych wiadomości otworzy okno(jeśli nie istnieje) i będzie zapisywał nowe wiadomości do odpowiednich zakładek.

Rysunek 33. Rozmowa pomiędzy użytkownikami Admin oraz Robert. Źródło opracowanie własne.

Struktura aplikacji serwera

Serwer jest to aplikacja, która przyjmuje żądania od klientów i wysyła je odpowiednio do innych klientów lub przetwarza i zwraca odpowiedź. Oprogramowanie serwera zostało napisane, jako oprogramowanie współbieżne. Każdy klient, który łączy się z serwerem jest traktowany, jako osobny wątek.



Rysunek 34. Struktura oprogramowania serwera. Źródło: opracowanie własne.

Klasy *AsymmetricEncryption*, *Encryption*, *EncryptionAES*, *EncryptionRSA*, są identyczne jak w oprogramowaniu klienta, zostaną pominięte w tym rozdziale. Opis zacznę od bardzo ważnej klasy *Client*. Klasa ta implementuje interfejs *Runnable*, który służy do tworzenia wątków w Java.

```
public class Client implements Runnable
{
    protected Socket connection;
    protected Protocol protocol;
    protected Db db;
    protected String ip;
    protected java.io.BufferedInputStream inBuffered;
    protected java.io.BufferedOutputStream outBuffered;
    protected java.io.DataInputStream in;
    protected java.io.DataOutputStream out;
    protected String line = "";
    protected String outLine = "";
    protected byte[] buffer;
    protected User user;
    protected AsymmetricEncryption asEncryption;
    protected EncryptionAES sEncryption;
```

Rysunek 35. Deklaracja i składowe klasy *Client*. Źródło: opracowanie własne.

Składowa *Socket* jest to gniazdo komunikacyjne służące do połączenia z klientem. *Protocol* jest to klasa, która obsługuje protokół komunikacyjny pomiędzy serwerem, a klientem. Klasa Db, jest to klasa odpowiedzialna za połączenie z bazą danych MySQL. Połączenie z bazą jest wymagane do autoryzacji użytkownika na podstawie konta utworzonego w bazie danych. Ip jest to obiekt, który przechowuje w postaci łańcucha znaków, numer IP klienta połączonego z serwerem. Klasy *BufferedInputStream*, *BufferedOutputStream*, *DataInputStream*, *DataOutputStream* są to klasy służące do pisania lub czytania z gniazda komunikacyjnego. Obiekty line oraz outLine są to zmienne pomocnicze przechowujące odpowiedzi z serwera. Wskaźnik do tablicy bajtów buffer służy do odbioru wiadomości od klienta w postaci bajtowej. Obiekt user, jest to obiekt klasy *User* służący do przechowywania informacji o użytkowniku. AsEncryption oraz sEncryption są to obiekty przechowujące kolejno szyfrowanie asymetryczne i symetryczne. Klasa Client posiada dwie metody: getUser() oraz run(). Metoda getUser() zwraca użytkownika, który jest przypisany do wątku. Metoda run() jest to wymagana przez interfejs *Runnable* metoda, która odpowiedzialna jest za komunikacje z serwerem. Metoda ta negocjuje klucz symetryczny z użytkownikiem. Następnie po negocjacji klucza, rozpoczyna ona

nieskończoną pętle, która ma za zadanie nasłuchiwać komendy od klienta i odpowiednio je obsługiwać.

Rysunek 36. Kod programu serwera obsługujący negocjację klucza. Źródło: opracowanie własne.

Kolejną klasą, którą opiszę jest klasa *Db.* Służy ona do połączenia się z bazą danych.

Rysunek 37. Struktura klasy *Db*. Źródło: opracowanie własne.

Klasa ta do połączenia z bazą potrzebuje connectora[19]. Po dodaniu do projektu biblioteki connectora, klasa ta tworzy nowy obiekt properties gdzie trzymane są dane niezbędne do połączenia z bazą danych MySQL, takie jak login, hasło, adres IP bazy

[19] Connector jest to biblioteka służąca do połączenia z bazą danych, dostarczana przez producenta bazy. W naszym przypadku jest to firma Oracle, a connector może zostać pobrany ze strony http://www.mysql.com/downloads/connector/

oraz nazwę bazy danych. Klasa implementuje wzorzec projektowy Singleton, przez co dostęp do niej jest otrzymywany poprzez wywołanie statycznej metody getInstance(). Metoda connect() służy do połączenia się z bazą danych. Metoda reconnect() jest używana do ponownego połączenia się z bazą.

Klasa *MessageXML* jest to klasa, która służy do tworzenia obiektów XML, które przechowują wiadomości w postaci znaczników.

Rysunek 38. Wiadomość w postaci XML. Źródło: opracowanie własne.

Wiadomość zawiera pola:

- from – jest to pole służące do przechowywania informacji o identyfikatorze użytkownika, który wysłał wiadomość,
- fromName – jest to nazwa użytkownika, który wysłał wiadomość,
- to – jest to numer indentyfikacyjny użytkownika, do którego kierowana jest wiadomość,
- toName – jest to nazwa użytkownika, do którego jest wysyłana wiadomość,
- timestamp – to znacznik czasu w postaci sekund, które upłynęły od 1 stycznia 1970 0:00 UTC,
- content – zawiera treść wiadomości do użytkownika.

Klasa *MessagesHandler* służy do przechowywania wiadomości użytkowników oraz do ich konwersji na format XML. Klasa ta domyślnie przechowuje w postaci list 100(liczba wiadomości jest ustalana poprzez parametr MAX_MESSAGE_COUNT) ostatnich

wiadomości i w przypadku przekroczenia tej liczby, usuwa najstarszą wiadomość i dodaje nową na początek listy.

Rysunek 39. Struktura klasy *MessagesHandler*. Źródło: opracowanie własne.

Zakończenie

Celem pracy było przedstawienie czytelnikowi wielu aspektów bezpieczeństwa komunikacji internetowej na praktycznym przykładzie bezpiecznego komunikatora internetowego. Praca przedstawia badania pokazujące, iż problem bezpieczeństwa komunikacji i danych w firmach istnieje i jest dość poważny. W pracy zostały przedstawione badania, które potwierdzają tą tezę. Począwszy od pierwszego rozdziału starałem się pokazać czytelnikowi, standardy bezpieczeństwa internetowego, zagrożenia na podstawie realnych danych, które występują w sieci. Typy różnych ataków komputerowych, które mogą być groźne dla bezpieczeństwa komunikacji, przedstawiłem również techniczne aspekty tychże ataków. Czytelnik mógł zapoznać się z algorytmami szyfrowania począwszy od tych prostych służących zrozumieniu tematu, do tych stosowanych w praktyce. Przedstawiłem również konkurencyjne produkty istniejące na rynku, które odpowiadają za bezpieczeństwo komunikacji internetowej. Stworzyłem aplikację, która jest bezpiecznym programem służącym do komunikacji, mającym zastosowanie w komercyjnej firmie, gdy istnieje potrzeba stworzenia bezpiecznego kanału komunikacyjnego. Uważam, iż praca w sposób bardzo dobry przedstawia najważniejsze zagadnienia związane z tematem bezpieczeństwa komunikacyjnego oraz pokazuje w nawiązaniu do wcześniejszych rozdziałów, jak napisać aplikację, która uchroni nas przed nieautoryzowanym dostępem do poufnych informacji. W pracy zapoznałem czytelnika z różnymi punktami koniecznymi do zrozumienia bezpieczeństwa informacji. Uważam, iż cel pracy został w pełni osiągnięty.

Bibliografia

[1] Niels Ferguson, Bruce Schneier, *Kryptografia w praktyce*, Helion, Gliwice 2004.

[2] Marcin Karbowski, *Podstawy kryptografii*, Helion, Gliwice 2007.

[3] Marcin Lis, *Java*, Helion, Gliwice 2002.

[4] Bruce Eckel, *Thinking in java*, Helion, Gliwice 2006.

[5] William Pollock, *Hacking: the art of exploitation, 2nd edition*, No Starch Press, Inc. 2008

[6] William Stallings, *Cryptography and Network Security Principles and Practice Fifth Edition*, Pearson Education Inc. 2011

[7] Stuart M., Joel S. and George K., *Hacking Exposed 7: Network Security Secrets & Solutions, Seventh Edition*, The McGraw-Hill Companies 2012

[8] Dorothy Denning, *Wojna informacyjna i bezpieczeństwo informacji*, Wydawnictwa Naukowo-Techniczne 2002

[9] Kenneth L. Calvert, Michael J. Donahoo, *TCP/IP Sockets in Java Practical Guide for Programmers Second Edition*, Elsevier Inc. 2008

[10] http://www.pwc.co.uk

[11] http://pl.wikipedia.org/

[12] http://www.iso27000.pl/sites/view/site=85

[13] http://www.123edi.com/edi-tdcc-101.asp

[14] http://www.mysql.com

[15] http://www.java.com/pl

Spis rysunków

Spis tabel