

Bezpieczeństwo Voice over IP opartego na SIP

Wojciech Mazurczyk

Instytut Telekomunikacji
Politechnika Warszawska
E-mail: W.Mazurczyk@elka.pw.edu.pl
<http://security.tele.pw.edu.pl/>

Streszczenie

W artykule opisano protokół SIP (Session Initiation Protocol) jako najbardziej obiecujący protokół sygnalizacyjny dla realizacji usługi Voice over IP. Szczególny nacisk położono na przedstawienie i analizę mechanizmów bezpieczeństwa zastosowanych w dwóch zaleceniach organizacji IETF dla SIP: RFC 2543 oraz RFC 3261. Zaprezentowano przyjęte w analizie kryterium oceny mechanizmów bezpieczeństwa oraz wskazano słabe punkty architektury bezpieczeństwa SIP w obu wymienionych zaleceniach. Dodatkowo przedstawiono przeprowadzone badania praktyczne oraz uzyskane wyniki wybranych aplikacji będących implementacjami Agentów Użytkownika SIP.

1 Wstęp

Kwestia bezpieczeństwa w obecnym świecie nabiera coraz większego znaczenia. Jest to pojęcie bardzo szerokie pojawiające się w wielu aspektach naszego życia. Wydźwięk tego słowa jest zawsze pozytywny – każdy chciałby by wszystko wokół niego było bezpieczne lub chce się czuć bezpieczny.

Potrzeba zapewniania bezpieczeństwa nie ominęła również obszaru teleinformatyki. Wraz ze wzrostem znaczenia informacji oraz biorąc pod uwagę dodatkowo fakt, iż obecnym trendem jest dążenie współczesnych społeczeństw do informatyzacji życia codziennego (społeczeństwo informacyjne) wymaga się, aby informacja docierała tylko i wyłącznie do właściwych adresatów i aby w czasie transportu nie uległa przekłamaniu lub podsłuchaniu. Jednak czy jesteśmy absolutnie przekonani, że oferowane nam rozwiązania teleinformatyczne (oprogramowanie, systemy itp.) są całkowicie bezpieczne, a może jesteśmy za bardzo dociekliwi?

Niniejszy artykuł poświęcony jest problemowi bezpieczeństwa usługi Voice over IP bazującej na protokole SIP. W świetle powyższych faktów bezpieczeństwo jest niewątpliwie krytycznym obszarem dla realizacji tego typu usługi.

2 Cel artykułu

Celem niniejszego artykułu jest przedstawienie, analiza i ocena trafności wyboru zastosowanych mechanizmów bezpieczeństwa protokołu SIP dla dwóch zaleceń:

- SIP w wersji pierwszej - RFC 2543 (rok opracowania 1999)
- SIP w wersji drugiej - RFC 3261 (rok opracowania 2002)

Przedstawiono również wyniki badań przeprowadzonych za pomocą zrealizowanego narzędzia pomocniczego do testowania bezpieczeństwa aplikacji będących implementacjami Agentów Użytkownika SIP (SIP UA) po to, aby można było dokonać sprawdzenia faktycznego stanu zabezpieczeń wykorzystywanego programu. Dodatkowo stworzono łatwo rozszerzalne testy, aby ułatwić testującemu weryfikację stanu zabezpieczeń badanych aplikacji.

Zastosowań dla omawianego protokołu powstało dotychczas wiele. W tej pracy rozważam bezpieczeństwo protokołu SIP **tylko** pod kątem możliwości oferowania bezpiecznych połączeń telefonicznych z wykorzystaniem jakże popularnych w naszych czasach sieci bazujących na protokole IP.

W artykule omówiony został protokół SIP w dwóch dostępnych wersjach mimo tego, że obowiązującym obecnie zaleceniem tego protokołu jest RFC 3261 (wersja druga). Jest to podyktowane faktem, iż wszystkie dostępne mi, bezpłatne aplikacje SIP UA wykorzystane w części praktycznej bazują tylko na RFC 2543 (spośród sześciu testowanych wszystkie oparte były na tej wersji).

3 Usługa VoIP

3.1 Wprowadzenie do VoIP

Technologia VoIP (Voice over Internet Protocol) nazywana również „telefonią internetową” lub „telefonią IP” oznacza przesyłanie głosu w czasie rzeczywistym wykorzystując do tego celu pakietową sieć transmisji danych z protokołem IP.

Umożliwia ona prowadzenie rozmów w sieciach lokalnych lub rozległych (np. Internecie), pozwalając na komunikowanie się z osobami lub instytucjami w praktycznie dowolnym miejscu na świecie korzystając z przyłączonego do danej sieci komputera bądź aparatu telefonicznego.

3.2 Historia telefonii IP

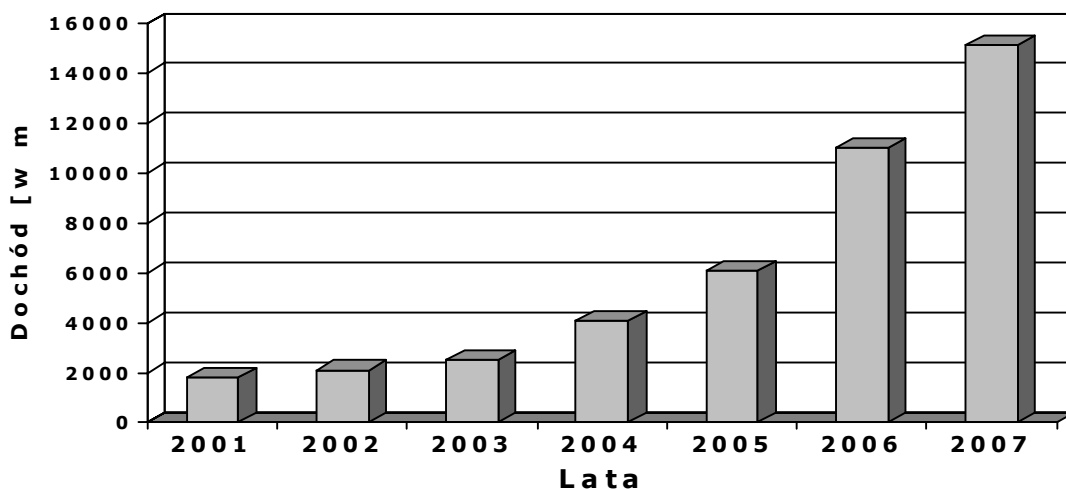
Historia tej techniki, co ciekawe, sięga lat siedemdziesiątych – pierwsze publikacje związane z transmisją głosu w sieciach pakietowych opublikowano już w RFC 741 (Specifications for the Network Voice Protocol) w 1977 roku. W tych czasach, jak również przez następne prawie 20 lat, jednak nikt poważnie nie traktował pomysłu telefonii IP – była to raczej ciekawostka pozostająca w kręgu zainteresowań badaczy i hobbystów. Wynikało to z braku rozwiązań zarówno technologicznych jak i ideowych (tj. braku odpowiednich protokołów i zbyt wysokich cen potrzebnych do realizacji takich rozwiązań procesorów sygnałowych DSP). Dopiero pod koniec XX w. jej rola znacznie wzrosła. Miało to związek głównie z niezaprzeczalnym sukcesem Internetu i nierozdzielnie z nim związanym protokołem IP.

Pierwszy system telefonowania IP został opracowany dopiero przez firmę VocalTec w 1995r., a pierwsze urządzenie VoIP (brama) weszła na rynek rok później. To właśnie wtedy użytkownicy mogli po raz pierwszy porozmawiać ze sobą w czasie rzeczywistym, używając do tego celu oprogramowania IP, które przesyłało przez sieć LAN pakiety zawierające dane audio.

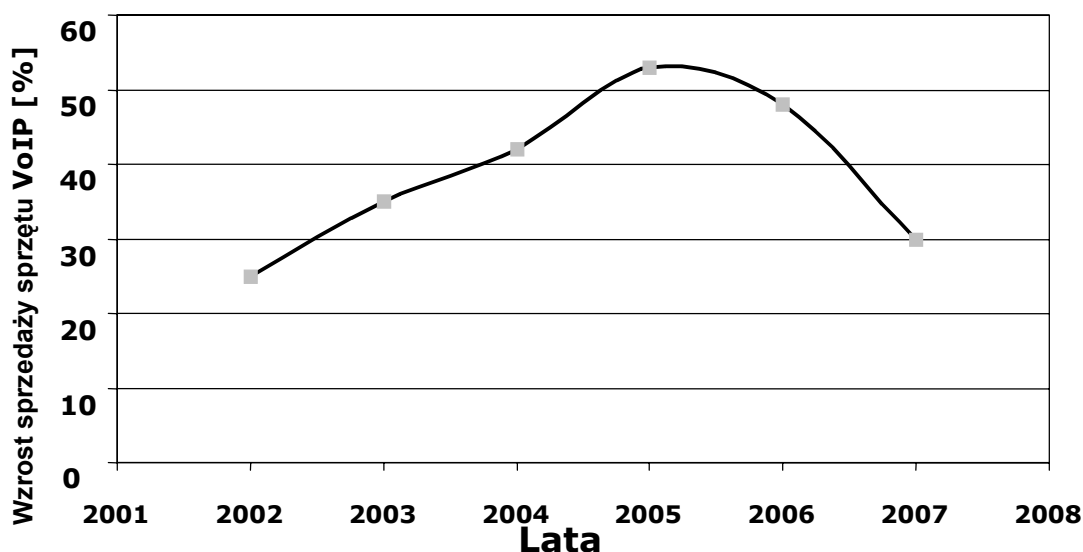
Szczególnie nęcącą wizją była i jest konwergencja sieci danych i telefonii w jedną uniwersalną sieć teleinformatyczną, co w wyniku może prowadzić do znacznych oszczędności głównie w firmach (wykorzystanie tej samej infrastruktury sieciowej).

3.3 Motywacja do zajęcia się VoIP. Perspektywy rozwoju

Według najnowszych badań opublikowanych w styczniu 2003 roku przez IDC (będącą jedną z największych firm zajmujących się badaniem rynku IT, twórcą wielu analiz sektorowych oraz strategicznym partnerem doradczym w wielu projektach - głównym celem jej działalności jest dostarczanie analiz na temat przyszłych trendów rozwojowych poszczególnych sektorów rynku) w pracy "Worldwide IP Telephony Equipment Forecast 2002-2007" prognozuje się 66% wzrost sprzedaży urządzeń VoIP dla firm w 2003 roku, co obrazuje obrazek Rys. 1.



Rys. 1. Prognozowany światowy dochód ze sprzedaży sprzętu dla VoIP (w milionach dolarów) w latach 2001-2007 (źródło: IDC)



Rys. 2. Prognozowany światowy wzrost (w procentach) sprzedaży sprzętu dla VoIP w latach 2001-2007 (źródło: IDC)

Również długotrwałe przewidywania dla rozwoju tej usługi są bardzo optymistyczne pomimo recesji na rynku telekomunikacyjnym. Szacunek IDC to roczny wzrost przychodów o 45% każdego roku aż do osiągnięcia 15,1 miliardów dolarów w roku 2007 (Rys. 2.). Wskazuje się na rok 2003 jako przełomowy w zdobyciu światowej popularności przez VoIP.

W świetle powyższych faktów przyczynami zainteresowania VoIP jest:

- Możliwość integracji sieci danych z „głosowymi” tak, aby móc zaspokoić wzrastające zapotrzebowanie na produkty wielousługowe (np. głos, dane i video),
- Konsolidacja pasma poprzez używanie jednej sieci dla wszystkich rodzajów ruchu,
- Olbrzymia popularność protokołu IP obsługiwane przez większość komputerów PC i stacji roboczych,
- Wzrost zapotrzebowania na złożone usługi np. czasu rzeczywistego, czy video typu „full motion”.

VoIP na pewno ma szansę stać się techniką, która może mieć ogromny wpływ na przyszły kształt całego rynku telekomunikacyjnego. Musi spełniać jednak kilka podstawowych warunków. Na początek powinna oferować podobny poziom i rodzaj usług za niższą cenę w porównaniu z tradycyjną telefonią – to może stanowić o jej powszechności. Poza tym musi dodatkowo zachęcać możliwościami, których nie posiada „zwykły” telefon, aby był sens dla indywidualnych użytkowników do zmiany operatora lub w przypadku firm, budowania własnych konwergentnych sieci IP. Komutowana sieć telefoniczna PSTN już zapewnia: oczekiwanie na połączenie, telekonferencje i przenoszenie rozmów – niedostępne jest tu jednak to, co może zaoferować przekaz pakietowy.

Dla przykładu:

- Ponad dziesięciokrotne zmniejszenie zapotrzebowania na pasmo (dzięki skutecznej kompresji cyfrowej głosu),
- Wsparcie dla przyszłych interakcyjnych funkcji multimedialnych (np. obraz ruchomy),
- Jednolite zarządzanie całością ruchu komunikacyjnego (dane, głos i obraz),
- Zapewnienie elastyczności komunikowania się.

Z powyższych faktów można wnioskować, że potencjał, który może „skusić” abonentów na pewno istnieje – pytanie tylko czy da się to wszystko w satysfakcjonujący sposób zaimplementować – innymi słowy, czy powstaną odpowiednie protokoły i urządzenia gwarantujące obiecane usługi?

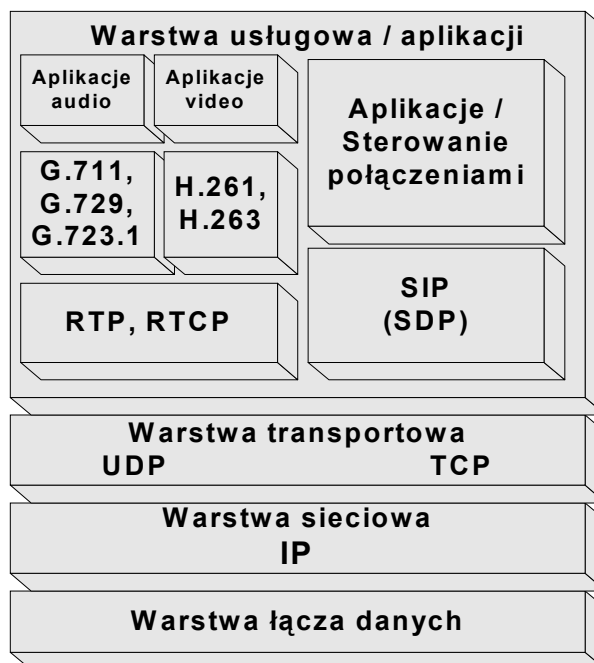
3.4 Protokoły umożliwiające „telefonię IP”

Całość sesji połączeniowej podzielona jest na dwie fazy: sygnalizacyjną (która kontroluje sesję) oraz transportującą media (tzn. strumień danych). Połączeniem w tym przypadku nazywamy, więc jedynie określony stan sygnalizacji pomiędzy urządzeniami końcowymi (a nie alokację fizycznego kanału).

Inną cechą charakterystyczną dla tradycyjnej telefonii jest fakt, iż podstawowym protokołem sygnalizacyjnym jest SS7 (Signalling System 7), który realizuje większość funkcji związanych z obsługą połączeń. W VoIP te same działania są realizowane przez **zespół protokołów**, który można podzielić na kilka grup:

- 1) Kodeki mowy (np. G.729),
- 2) Protokoły transportowe (RTP, UDP, TCP),
- 3) Protokoły sygnalizacyjne (H.323, SIP, MGCP, H.248/Megaco),
- 4) Protokoły uzupełniające (SDP, RTSP, RSVP).

Stos protokołów dla usługi VoIP bazującej na SIP (model TCP/IP) jest następujący:



Rys. 3. Stos protokołów dla SIP (model TCP/IP)

3.5 Protokoły sygnalizacyjne

Protokoły tej grupy są „sercem” telefonii internetowej, ponieważ od niej zależy architektura sieci, realizacja zaawansowanych usług oraz współdziałanie z tradycyjnymi sieciami telefonicznymi. W ogólności do podstawowych funkcji protokołu sygnalizacyjnego można zaliczyć:

- Translację adresów i lokalizację użytkownika wywoływanego,
- Otwieranie i zamykanie sesji,
- Negocjacje parametrów sesji i ewentualnie ich modyfikację w trakcie połączenia,
- Zarządzanie grupą uczestników sesji,
- Zarządzanie przebiegiem połączeń.

Niestety obecnie nie ma jednego międzynarodowego standardu protokołu sygnalizacyjnego dla VoIP (co jest kolejnym czynnikiem hamującym popularność tej techniki). Najbardziej znanymi protokołami tego rodzaju są: **SIP (Session Initiation Protocol)**, **H.323**, **MGCP (Media Gateway Control Protocol)** oraz **H.248/Megaco**.

3.6 Motywacja do zajęcia się protokołem SIP

Z wszystkich dostępnych protokołów sygnalizacyjnych dla telefonii IP to właśnie SIP wydaje się być najbardziej perspektywicznym i właśnie z nim wiąże się największe nadzieje.

Na dowód tego stwierdzenia przytoczę wyniki ankiety przeprowadzonej przez jedną z najbardziej liczących się niezależnych firm zajmującą się analizą i badaniem zarówno sprzętu jak i rynku IT: Mier Communication Inc. Przeprowadzona została ona wśród 150 najbardziej znanych producentów i sprzedawców sprzętu oraz oprogramowania VoIP. Wykazuje, iż mimo obecnej dominacji protokołu H.323, przyszłe produkty tych firm będą związane z wykorzystaniem protokołu SIP. Wiadomo, że błędnym założeniem byłoby na podstawie przyszłych preferencji ankietowanych firm

stwierdzić, iż Session Initiation Protocol zdominuje rynek protokołów sygnalizacyjnych dla telefonii internetowej. Już niejednokrotnie przecież w historii teleinformatyki zdarzało się tak, że wiele potencjalnie przyszłościowych rozwiązań zostało porzucanych pomimo tego, iż wcześniejsze prognozy stanowiły inaczej. Dodatkowymi argumentami za SIP stanowią jego niewątpliwe zalety, które postaram się przedstawić poniżej.

Protokół SIP (Session Initiation Protocol) jest tekstowym protokołem sygnalizacyjnym warstwy aplikacji umożliwiającym ustanawianie, modyfikację, likwidację oraz zarządzanie przebiegiem sesji, w których uczestniczy dwóch lub więcej użytkowników. Został on stworzony przez organizację IETF MMUSIC (IETF Multiparty Multimedia Session Control). Sesje SIP mogą być: internetowymi konferencjami multimedialnymi (telefonicznymi) lub multimedialną transmisją rozsiewczą. Wspierany jest zarówno tryb rozgłoszeniowy (multicast) jak również punkt-punkt (unicast).

Charakterystyczną cechą dla protokołu SIP odróżniającą go od innych protokołów sygnalizacyjnych wykorzystywanych dla VoIP jest zdolność do współpracy z użytkownikami zmieniającymi swoją lokalizację poprzez wykorzystanie żądań REGISTER i serwerów registrar. Dodatkowo jest ona niezależna od terminala użytkownika (telefonu, komputera itp.) – jest przypisana konkretnemu użytkownikowi. Dlatego też SIP został wybrany przez organizację 3GPP (Third Generation Partnership Project) protokołem dla aplikacji multimedialnych w sieciach trzeciej generacji UMTS.

Zasada działania tego protokołu, jak również sama organizacja (w tym mechanizmy bezpieczeństwa), są bardzo podobne do tych zawartych w dobrze znanym protokole HTTP (Hyper Text Transfer Protocol, którego twórcą jest również organizacja IETF) – w nagłówkach wiadomości SIP znajdują się takie same pola, jak w HTTP, co znacznie ułatwia jego integrację z serwerami WWW. Poza tym SIP może współpracować praktycznie z każdym protokołem transportowym, czy to połączeniowym (np. TCP), czy datagramowym (np. UDP). Dodatkowo zapewnia pełną integrację z otwartymi standardami i technologiami internetowymi. Wykorzystuje m.in. URI (Universal Resource Indicator), DNS (Domain Name Server), czy MIME (Multipurpose Internet Mail Extensions) w taki sposób, że są one kompatybilne z innymi aplikacjami IP. To pozwala temu protokołowi łatwo współpracować m.in. z aplikacjami Web.

Innymi zaletami tego protokołu są: stosunkowo mała złożoność oraz fakt, iż bazuje on na znanym modelu klient-serwer. Poza tym SIP został tak zaprojektowany, by być łatwo rozszerzalnym, co oznacza możliwość obsługi ciągle nowych wiadomości oraz typów usług. Przykładem może być tu rozszerzenie SIP (nie związane z telefonią IP) zwane SIMPLE (SIP for Instant Messaging and Presence) służące komunikatorom 'instant messaging'.

Koleją cechą jest kompatybilność wstecz. W momencie nie obsługiwanego przez jedną ze stron końcowych określonych rozszerzeń protokołu możliwe jest uzgodnienie ignorowania tychże i użycie „bazowego” protokołu do kontynuowania komunikacji.

Jednak doświadczenie ITU oraz poparcie firm produkujących sprzęt i oprogramowanie spowodowało wypromowanie H.323, jako obecnie podstawowego protokołu sygnalizacyjnego dla VoIP, co nie oznacza, że jest on najlepszym z istniejących.

Obecnie można już zaobserwować zmianę „trendu H.323” czego najlepszym przykładem może być rozległa oferta urządzeń firmy Cisco (AS5300, 2600 Series, 3600 Series, AS5400 universal gateway, AS5350 universal gateway oraz AS5850 universal gateway) obsługujących protokół SIP.

Dużą szansą dla SIP może też stanowić jego implementacja w najnowszym produkcie Microsoftu: Windows XP w aplikacji Windows Messenger. Dzięki temu ma on szansę wyjść z cienia H.323. Sam *XP product manager* w tej firmie Imad Yanni stwierdził, że wspieranie protokołu H.323 jest nadal ważne ze względu na dużą liczbę klientów i producentów, którzy już w tego typu rozwiązania zainwestowali. Jednak to SIP – twierdzi Yanni – ma większy potencjał by wprowadzać nowe coraz bardziej innowacyjne formy komunikacji. Dlatego Microsoft w przyszłości zamierza rozwijać usługi związane z komunikacją w kierunku zorientowanym na SIP, a planuje się zaprzestania wspierania rozwiązań H.323. Pomimo tego, że tacy producenci jak Alcatel, Cisco i Siemens dążą do tego, aby rozwój szedł równomiernie w obu kierunkach sygnalizacyjnych to poparcie potentata, jakim jest Microsoft, może spowodować większe zainteresowanie i popularność protokołu SIP. Przyszłość pokaże jak będzie w rzeczywistości.

4 Podstawy SIP

4.1 Adresowanie

„Obiektami” adresowanymi w protokole SIP są użytkownicy, którzy należą do różnych domen (hostów). Ich identyfikacja odbywa się na podstawie analizy SIP URL postaci np. „sip://j.kowalski@tele.pw.edu.pl”. Stosowany format adre-

sów jest zbliżony do tego stosowanego w adresach e-mailowych np. **user@host** (gdzie *user* oznacza nazwę użytkownika lub numer telefonu, a *host* jest nazwą domeny, bramy lub adresem IP) co ułatwia identyfikację i zapewnia pomoc w dotarciu, bo nawet nie znając dokładnego formatu adresu można go zgadnąć np. na podstawie maila. Ciekawostką dotyczącą adresowania jest możliwość umieszczenia odnośnika (SIP-URL) na stronie WWW umożliwiającego bezpośrednio dotarcie do danego użytkownika (click-to-call).

4.2 Architektura funkcjonalna protokołu SIP

Na architekturę funkcjonalną składają się następujące komponenty:

- **Agent Użytkownika (User Agent)** będący systemem końcowym (zazwyczaj odpowiednio inteligentnym oprogramowaniem), działającym w imieniu użytkownika, który uczestniczy w połączeniu. Składa się on z dwóch części: **klienta** (User Agent Client) i **serwera** (User Agent Server). Klient wysyła **żądania** protokołu SIP, natomiast serwer wysyła **odpowiedzi** w imieniu użytkownika oraz odbiera żądania przesyłane do niego przez innych agentów. Bezpośrednia komunikacja agentów opiera się, więc na popularnym modelu klient/serwer.
- **Serwery Sieciowe (Network Servers)**, których podstawową funkcją jest translacja adresów i pośredniczenie w procesie odnajdywania użytkownika, do którego jest skierowane żądanie.

Istnieją dwa podstawowe rodzaje serwerów sieciowych w SIP: **proxy** i **redirect**.

- **Proxy**, po otrzymaniu żądania, odpowiedzialny jest za ustalenie adresu następnego serwera, do którego należy je skierować i przy użyciu odpowiednich mechanizmów jak np. DNS zostaje ono tam przesyłane (może być generowane nowe żądanie SIP). Odpowiedź na to żądanie będzie powracać tą samą drogą w odwrotnym kierunku. Rozróżnia się dwa rodzaje serwerów proxy: **stanowy** i **bezstanowy**.
- Proxy stanowy charakteryzuje się tym, że każde zgłoszenie do niego przychodzące jest zapamiętywane oraz tworzony jest dla niego osobny proces. Musi być stosowany w przypadku, gdy serwer „rozwidła” (forking proxy) wysyłanie wiadomości, gdy współpracuje z protokołem TCP oraz przy realizacji zaawansowanych usług. Proxy bezstanowy interpretuje każdą wiadomość oddzielnie, nie łącząc ich w sekwencje żądanie-odpowiedź.
- **Redirect**, po odebraniu żądania, zajmuje się wysyłaniem do agenta użytkownika odpowiedzi zawierających adres następnego serwera, z którym należy się skontaktować w celu poszukiwania właściwego serwera użytkownika końcowego (nie może inicjować swojego własnego żądania SIP). Serwer redirect jest zawsze bezstanowy.

Dodatkowo funkcjonalność architektury opisywanego protokołu jest wzbogacana serwerem **registrar** będącego serwerem akceptującym żądania REGISTER. Odbiera on zgłoszenia rejestracyjne agentów użytkownika zawierające informacje o obecnej lokalizacji użytkownika. Lokalizowany jest on zwykle wspólnie z serwerami: proxy lub redirect.

Żądania wysyłane przez klienta wywołują w serwerze odpowiednie metody. Po otrzymaniu i zinterpretowaniu żądania agent wysyła odpowiedź, która oznajmia sukces lub niepowodzenie danej operacji albo wskazuje postęp w realizacji wywoływanej metody.

4.3 Żądania (metody) stosowane w protokole SIP

Zostało wyspecyfikowanych sześć podstawowych żądań w tym protokole (wg zalecenia RFC 2543):

- **INVITE** – wskazuje, że użytkownik lub serwer został zaproszony do udziału w sesji (konferencji). Treść ciała wiadomości (message body) zawiera opis sesji (wykorzystując protokół SDP) i typ mediów, który ma być wykorzystany podczas rozmowy, jak również adres źródłowy i docelowy, lokalizację użytkownika oraz preferencje dzwoniącego. Może się tu również znaleźć propozycja wybranego przez dzwoniącego kodeka mowy. Żądanie musi być zaimplementowane w serwerach proxy, redirect oraz w agencie użytkownika,
- **ACK** – potwierdza, że klient otrzymał końcową odpowiedź na żądanie INVITE. Służy do zapewnienia niezawodnej wymiany INVITE (jest używane tylko z tą metodą). Serwer zawsze retransmituje odpowiedzi finalne

aż do uzyskania potwierdzenia ACK od klienta. W ciele wiadomości (message body) może zawierać informacje ostatecznego opisu sesji, z którego powinien korzystać dzwoniący w przeciwnym razie należy użyć parametrów zawartych w żądaniu INVITE. Obsługa tej metody wymaga się od agentów użytkowników oraz serwerów: proxy i redirect,

- **OPTIONS** – przekazuje informację o funkcjonalności (capabilities) – nie zestawia połączenia. Musi być obsługiwana przez serwery SIP: proxy i redirect, agenta oraz registrar,
- **BYE** – oznacza chęć zakończenia połączenia między dwoma użytkownikami sesji. Połączenie może być rozłączone zarówno przez użytkownika, który zainicjował połączenie jak i przez jego odbiorcę,
- **CANCEL** – przerywa żądanie będące w toku, które zawiera te same deskryptory połączenia, ale nie wpływa na już ukończone żądania,
- **REGISTER** – przynosi informację o lokalizacji użytkownika dla serwera SIP. Po wykonaniu rejestracji z adresem użytkownika jego lokalizacja zostanie skojarzona. Może ona się odbyć w dowolnym momencie. Protokół SIP zezwala klientowi na rejestrację z różnych lokalizacji – dla przykładu na początku agent użytkownika może wysłać IP ‘multicast’ do wszystkich aktywnych serwerów SIP w domenie (np. sip.mcast.net, 224.0.1.75). Standardowo użytkownicy posługujący się adresami SIP powinni posiadać swój serwer *registrar* w domenie, do której należą i w nim dokonywać rejestracji.

Funkcje, które nie są obsługiwane przez serwery: proxy i redirect są przez nie traktowane jako metody OPTIONS. Natomiast te, których nie zapewniają serwer agenta użytkownika oraz registrar powodują wysłanie odpowiedzi „nie zaimplementowane”.

4.4 Rodzaje odpowiedzi

Kody wykorzystane do przesłania statusu (Status-Codes) obecnie wykonywanego żądania zostały podzielone na następujące klasy:

1xx - klasa **Informująca** (Informational)– żądanie zostało odebrane, podjęto kroki w celu jego wykonania, nie napotkano żadnych problemów - wskazuje na postęp w wywoływaniu metody, zawsze po tego typu kodzie następują wiadomości określające status zakończonej operacji.

2xx – klasa **Sukces** (Success) – akcja została zakończona sukcesem.

3xx – klasa **Przekierowanie** (Redirection) – dalsze czynności muszą być podjęte w celu wypełnienia żądania.

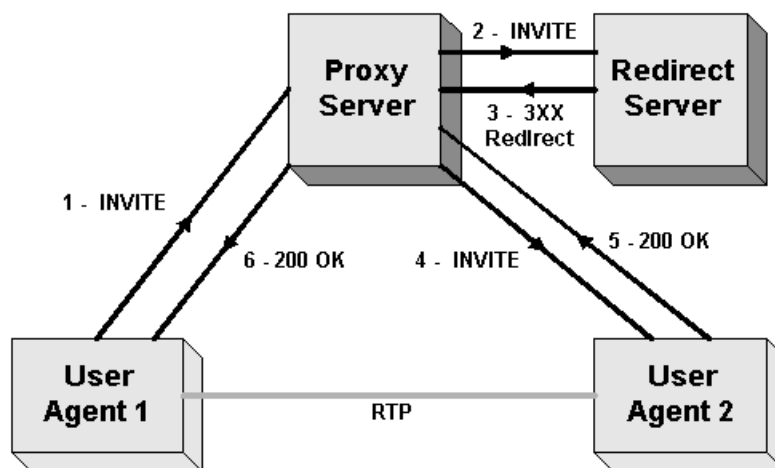
4xx – klasa **Błąd Klienta** (Client Error) – żądanie zostało błędnie sformułowane lub nie może być wypełnione przez dany serwer i zostanie odrzucone.

5xx – klasa **Błąd Serwera** (Server Error) – serwer nie zdołał wypełnić pozornie poprawnie sformułowanego żądania.

6xx – klasa **Błąd Globalny** (Global Failure) – żądanie nie może być wypełnione przez żaden z dostępnych serwerów.

4.5 Model inicjacji połączenia

Ogólna zasada nawiązania połączenia, w której występują wszystkie elementy architektury funkcjonalnej protokołu SIP, została przedstawiona na rysunku poniżej:



Rys. 4. Połączenie nawiązane za pomocą metody INVITE

Analiza przebiegu połączenia:

1. Od użytkownika (User Agent 1) zostaje wysłane żądanie INVITE (1), które ma zainicjować połączenie z odbiorcą (User Agent 2)
W wiadomości zostają wyszczególnione: adres SIP-URL nadawcy (pole From) i odbiorcy (pole To) oraz zostaje nadany unikalny identyfikator połączenia Call-ID, który wskazuje bezpośrednio na serwer inicjującego.
2. Wiadomość INVITE zostaje odebrana przez sieciowy serwer proxy, który odpowiedzialny jest za ustalenie adresu następnego serwera, do którego należy je skierować.
3. Następnie wiadomość INVITE (2) zostaje przesłana do sieciowego serwera redirect, który zajmuje się wysyłaniem do proxy odpowiedzi zawierającej adres następnego serwera, z którym należy się skontaktować w poszukiwaniu właściwego serwera użytkownika końcowego.
4. W wyniku działania tegoż odnaleziona zostaje domena odbiorcy (3).
5. Następnie serwer proxy ponawia wysłanie żądania INVITE, tym razem używając otrzymanego ostatnio adresu (4).
6. Serwer, do którego została skierowana wiadomość odpowiada akceptując połączenie przesłane serwerowi docelowemu odpowiedzi 200 (OK).
7. Źródłowy serwer agenta użytkownika przesyła wiadomość potwierdzenia ACK do odbiorcy i następuje faza rozmowy (pakiety RTP).

5 Bezpieczeństwo połączeń VoIP

Zagwarantowanie bezpiecznych połączeń dla usługi VoIP jest sprawą złożoną. Nie ogranicza się ono tylko do zapewnienia zabezpieczonego transportu strumieni danych zawierających głos, ważniejszą sprawą jest fakt, w jakich warunkach przesyłane są wiadomości protokołu sygnalizacyjnego, na którym bazuje VoIP.

Problemy bezpieczeństwa dla usługi telefonii IP w świetle powyższych stwierdzeń można podzielić na:

- a. Bezpieczeństwo wiadomości sygnalizacyjnych wymienianych pomiędzy stronami komunikującymi się,
- b. Bezpieczeństwo pakietów przynoszących głos (pakiety RTP),
- c. Problemy związane z „przechodzeniem” pakietów przez ściany przeciwogniowe (Firewalls) oraz przez mechanizmy translacji adresów wewnętrznych (np. intranetowych) na zewnętrzne (np. internetowe) NATs (Network Address Translators).

Niniejszy artykuł skupia się wyłącznie na tematyce zawartej w punkcie a; bezpieczeństwo przepływu pakietów „z głosem” (b) jest zapewniane niezależnie od gwarantowania bezpiecznej sygnalizacji. Trzecia grupa problemów (c) nie jest związana bezpośrednio z bezpieczeństwem samego protokołu SIP – dotyczy raczej sieci będącej implementacją usługi VoIP opartej na protokole SIP. W takiej sieci z powodu wymienionych „przeszkód” może dojść do uniemożliwienia:

- Nawiązania połączenia z powodu blokowania pakietów zawierających sygnalizację SIP,
- Porozumiewania się przez strony żądające komunikacji ze względu na blokowanie pakietów RTP.

6 Aspekty bezpieczeństwa w SIP

W czasie fazy sygnalizacyjnej określone parametry sesji są wymieniane pomiędzy użytkownikami końcowymi w celu poprawnej realizacji żądanego połączenia. Mogą one zawierać informacje, które użytkownik wolałby pozostawić niedostępne dla osób trzecich (np. jego lokalizacja, czy nazwisko). Ważne jest również to, aby użytkownicy, którzy nie są uwierzytelnieni nie mieli możliwości zmiany, wstawiania oraz usuwania wiadomości wysyłanych w czasie fazy sygnalizacyjnej. Aby zapewnić te oraz inne aspekty bezpieczeństwa, które są już przy poziomie rozwoju teleinformatyki konieczne dla każdej nowoczesnej sieci stosuje się mechanizmy takie jak np. szyfrowanie, uwierzytelnienie, funkcje skrótu oraz podpisy cyfrowe.

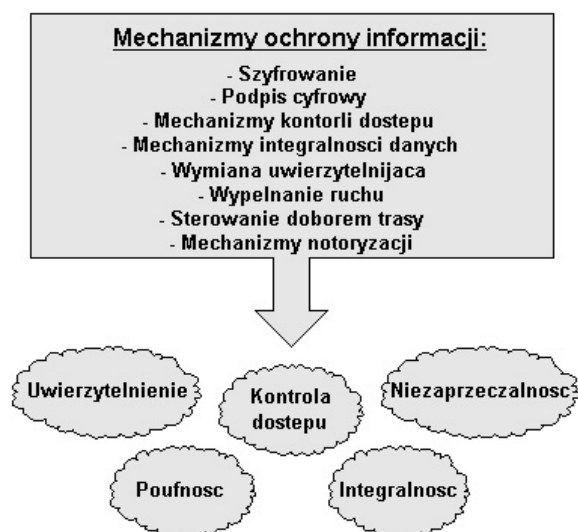
6.1 Problem doboru kryterium oceny bezpieczeństwa dla SIP

Aby usystematyzować analizę mechanizmów bezpieczeństwa protokołu SIP trzeba na początku ustalić kryterium, według którego nastąpi jego analiza. Nie jest to wbrew pozorom prosta sprawa, gdyż należy przy tym wyborze kierować się specyfiką protokołu, charakterystyką potencjalnych zagrożeń itp. tak, aby być w stanie rozważyć wszystkie aspekty zapewniania bezpieczeństwa jak najpełniej.

Przedstawiony poniżej wywód prezentuje tok rozumowania autora, który został przyjęty przy ustalaniu niezbędnego do dalszej analizy kryterium bezpieczeństwa protokołu SIP. Wymagało to przypomnienia kilku podstawowych definicji z zakresu ochrony informacji. Zaleca się, aby czytelnik posiadający tę wiedzę ograniczył się tylko do pobieżnego zapoznania z niniejszym punktem.

Spośród dostępnych, potencjalnych kryteriów zdecydowałem się wybrać rozwiązanie będące modyfikacją podziału zawartego w normie ISO 7498-2, według którego bezpieczeństwo w systemach otwartych należy rozpatrywać w kontekście możliwości zapewnienia pięciu podstawowych **usług** ochrony informacji (kontrola dostępu (access control), uwierzytelnienia (authentication), integralności danych (data integrity), poufności danych (confidentiality) oraz niezaprzeczalności (nonrepudation)).

Podstawowe usługi przedstawione powyżej są budowane na bazie **mechanizmów**. Wspomniana przeze mnie norma ISO definiuje je również. Przedstawia je poniższy rysunek:



Rys. 5. Mechanizmy budujące podstawowe usługi ochrony informacji

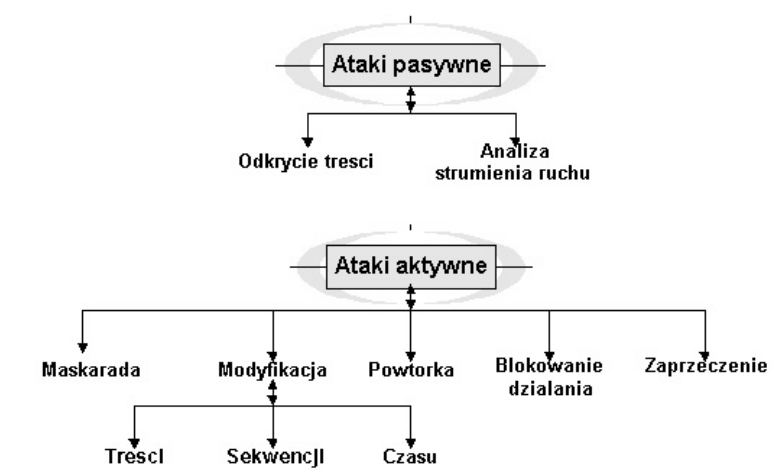
Postaram się teraz wykazać, które z wymienionych usług wpływają w sposób znaczący na bezpieczeństwo protokołu SP. Usługa integralności zawiera się w uwierzytelnieniu, a ono tworzy niezaprzeczalność z tą różnicą, że dla niezaprzeczalności dodatkowo nie ma możliwości wyparcia się przez nadawcę faktu wysłania określonej wiadomości. Zapewnienie tej usługi nie jest sprawą najistotniejszą w przypadku przesyłania wiadomości sygnalizacyjnych protokołu SIP, które kontrolują lub modyfikują sesje użytkowników, raczej liczy się to w przypadku np. transakcji finansowych, itp. Można jednak założyć, że podstawy niezaprzeczalności są obecne w SIP w postaci realizowania usługi uwierzytelnienia.

Natomiast usługa kontroli dostępu do zasobów nie odnosi się bezpośrednio do wymiany wiadomości sygnalizacyjnych SIP, bo przecież dostęp np. do usługi telefonicznej powinien mieć każdy, a to czy ktoś nieupoważniony nie narusza zasobów w sieci jest poza gwarancjami bezpieczeństwa oferowanymi przez omawiany tu protokół sygnalizacyjny. Poza tym uwierzytelnienie w sieci oparte na protokole SIP odbywa się w celu ustanowienia sesji lub uzyskania dostępu do określonych usług sieciowych, a zatem pośrednio realizowana jest usługa kontroli dostępu.

Zastosowane przeze mnie kryterium, aby być miarodajnym, powinno dodatkowo uwzględniać jak najwięcej klas zagrożeń oraz potencjalnych problemów bezpieczeństwa, ponieważ wymiana informacji pomiędzy uczestnikami sesji może podlegać szeregowi ataków. Zatem kryterium powinno zostać dopasowane tak, aby jak najpełniej oddać również tą składową problemu.

Zagrożenia powstałe na skutek celowej akcji ze strony intruza mogą mieć charakter **pasywny** (istnieje jedynie możliwość podsłuchu lub stwierdzenia faktu przepływu wiadomości) lub **aktywny** (może dojść do ingerencji w przesyłanie wiadomości np. poprzez zmianę jej zawartości).

Podział na poszczególne **klasy ataków** na sieć oraz na komunikowanie się w niej z podziałem na pasywne i aktywne przedstawia poniższy rysunek. Klasyfikacja ta jest zmodyfikowaną wersją podziału wg Stallings'a.



Rys. 6. Podział ataków na sieć oraz komunikację w niej

Konkludując można przyjąć odpowiadające specyfice protokołu SIP kryterium oceny mechanizmów bezpieczeństwa tego protokołu jako umiejętność zapewnienia dwóch głównych usług ochrony informacji oraz komunikacji w sieci tzn.:

- **Poufności** – dającej ochronę przed atakami pasywnymi oraz zabezpieczającej wiadomości sygnalizacyjne wymieniane pomiędzy agentami użytkowników komunikującymi się ze sobą przed nieuprawnionym ich uzyskaniem przez strony do tego nieupoważnione;
- **Uwierzytelnienia** – gwarantującego ochronę przed większością ataków aktywnych oraz kontrolę tożsamości stron i/lub wiadomości sygnalizacyjnych wymienianych pomiędzy nimi. Podobnie jak w normie ISO 7498-2 integralność wiadomości zawiera się w uwierzytelnieniu, a ono następnie stanowi główną część niezaprzeczalności.

Dlatego przy omawianiu mechanizmów bezpieczeństwa protokołu SIP główny nacisk położę właśnie na sposób zapewnienia właśnie tych dwóch usług.

6.2 Techniki ataków na protokół SIP

W świetle przedstawionego podziału na klasy ataków na bezpieczeństwo sieci i komunikacji w niej oraz usług ochrony informacji, jakie powinny być oferowane postaram się zidentyfikować i sklasyfikować potencjalne techniki ataków na SIP - techniki, czyli określone działania oraz narzędzia użyte do ataku, które mogą być wykorzystane w różny sposób, a zatem jedna technika może być wykorzystana do stworzenia wielu klas zagrożeń (typów ataków). Poniżej w tabeli prezentuję najczęściej używane techniki.

Technika ataku:	Opis:	Klasy zagrożeń dla sieci i wiadomości w niej przesyłanych:
Podszycie się (Spoofing)	Technika ta polega na podszywaniu się pod cudze adresy IP dzięki czemu możliwe jest np. wysyłanie sfalszowanych informacji	Odkrycie treści wiadomości, Modyfikację, Maskaradę, Blokowanie działania, Powtórkę, Zaprzeczenie
Podsluchiwanie (Sniffing)	To technika monitorowania i analizowania pakietów (pasywne nasłuchiwanie) wszystkich wiadomości przesyłanych przez sieć.	Odkrycie treści wiadomości, Powtórka
Blokowanie działania (Denial of Service - DoS)*	Polega na blokowaniu funkcjonowania elementów sieci za pomocą wysyłania dużej liczby wiadomości. Tak obciążony element nie wytrzymuje naporu wiadomości i zostaje zablokowany.	Blokowanie działania
*) istnienie takiej techniki ataku nie wyklucza istnienia klasy ataków o takiej samej nazwie		

Tab. 1. Techniki wykorzystywane do ataków na protokół SIP

Przykłady typowych ataków z pomocą scharakteryzowanych technik to:

Podszycie (*Spoofing*) może powodować wiele klas zagrożeń i występuje w wielu postaciach. Przeprowadzenie ataku z wykorzystaniem tej techniki jest podobne do ataku na protokół SMTP. Dla protokołu SIP mogą to być sytuacje, gdy następuje *spoofing*:

- Pola *From* w metodzie REGISTER (przekierowanie połączenia) – zwane również *Registration Hijacking*,
- Pola *From* w metodzie INVITE.
- *Man in the Middle* – atakujący jest w stanie podsłuchać serwer proxy i również zmieniać kluczowe informacje w wiadomościach,
- *Impersonating a Server* – agent użytkownika kontaktuje się z serwerem proxy, w celu dostarczenia żądania, a intruz podszywa się pod serwer. Sytuacja szczególnie się komplikuje w przypadku obsługi użytkowników mobilnych,
- *Tearing down session* – wstawienie przez atakującego odpowiedzi BYE w czasie, gdy zachodzi komunikacja między użytkownikami.

Posłuchanie (Sniffing) też może przyjmować dla SIP następujące postacie:

- Podsluchiwanie wiadomości sygnalizacyjnych (*Eavesdropping*),
- Penetracja ciała wiadomości sygnalizacyjnych (*Tampering with message bodies*).

Istnienie wymienionych powyżej rodzajów ataków powoduje konieczność istnienia określonych rodzajów mechanizmów bezpieczeństwa koniecznych, aby im przeciwdziałać. Przejdźmy teraz do ich analizy.

7 Architektura bezpieczeństwa SIP w zaleceniu RFC 2543

Session Initiation Protocol nie definiuje ściśle określonego typu uwierzytelnienia ani techniki stosowanej do realizacji mechanizmu szyfrowania (potencjalnie łatwa rozszerzalność). Jakkolwiek proponuje jak można wykorzystać mechanizmy uwierzytelniające zastosowane w protokole HTTP, szyfrowanie oraz generację podpisów cyfrowych z użyciem PGP (Pretty Good Privacy).

7.1 Uwierzytelnienie

Uwierzytelnienie wiadomości jest to procedura sprawdzania, czy pochodzą one z zaznaczonego w nich źródła i czy nie zostały zmienione w czasie ich transportu (czyli integralność). Dodatkowo może ona obejmować weryfikację kolejności oraz autentyczności czasu.

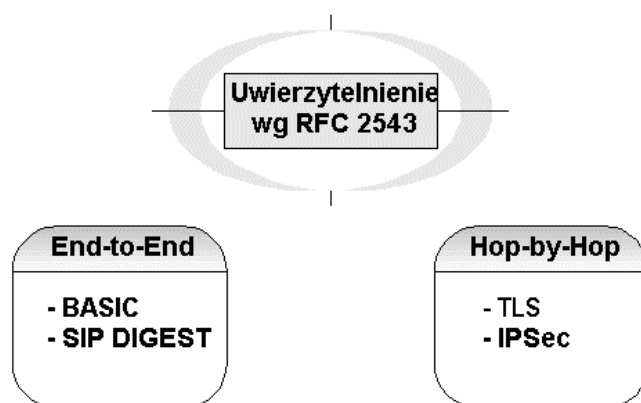
Usługa uwierzytelnienia jest możliwa z punktu widzenia kierunku jej wykonania jest dostępna w dwóch trybach: jednokierunkowym oraz dwukierunkowym (*mutual* - wzajemna).

Uwierzytelnienie w sieci opartej na protokole SIP jest procesem, w którym agent użytkownika dostarcza swoje dane uwierzytelniające serwerowi SIP (głównie proxy) lub innemu agentowi użytkownika w celu:

- Ustanowienia sesji
- Uzyskania dostępu do określonych usług sieciowych.

Mechanizmy uwierzytelniające opisane w zaleceniu RFC 2543 można podzielić ze względu na obszar realizacji omawianej usługi w odniesieniu do drogi komunikacyjnej na mechanizmy typu **End-to-End** (obsługa bezpośrednia źródło->cel) oraz **Hop-by-Hop** (obsługa tranzytowa – tylko pojedyncze połączenie warstwy transportowej). Podobny podział będzie również wprowadzony w przypadku zalecenia RFC 3261).

Wyszczególnienie konkretnych mechanizmów z uwzględnieniem tego podziału zostało umieszczone na rysunku poniżej.



Rys. 7. Realizacja usługi uwierzytelnienia w SIP wg RFC2543

W dalszej części tego rozdziału zostaną przedstawione wymienione mechanizmy z uwzględnieniem istniejącego na rysunku podziału.

7.2 Uwierzytelnienie typu End-to-End

Aby zapewnić tę usługę protokół SIP oferuje dwa różne mechanizmy: *Basic* i *Digest*. Nie były one specjalnie zaprojektowane dla tego protokołu sygnalizacyjnego – zostały one zaczerpnięte w prawie niezmienionej formie z protokołu HTTP.

Oba mechanizmy bazują na wykorzystaniu czterech pól nagłówka przesyłanej wiadomości, z których część jest charakterystyczna dla serwerów proxy, a reszta dla agenta użytkownika. Są to: *WWW-Authenticate*, *Authorization*, *Proxy-Authenticate* oraz *Proxy-Authorization* (oraz dla *Digest* dodatkowo może być użyte pole: *Authentication-Info*).

7.3 Uwierzytelnienie Basic

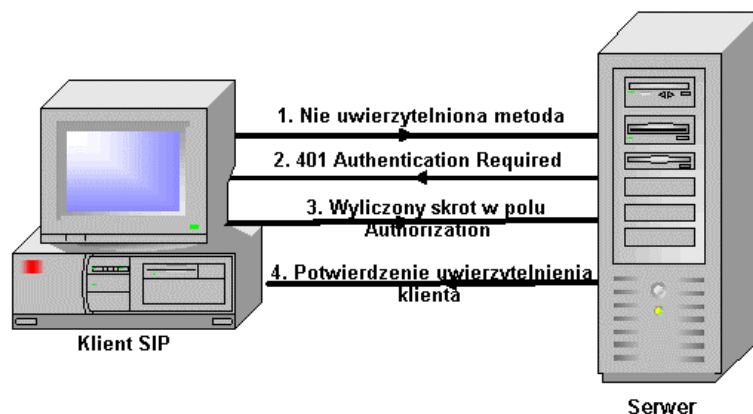
Bazuje na schemacie współdzielonego sekretu (shared secret) - użyty klucz jest znany przez obie strony (serwera i klienta). Mechanizm ten jest jednak bardzo prymitywny i nie gwarantuje wystarczającego poziomu bezpieczeństwa, ponieważ dane uwierzytelniające (hasło) użytkownika są przesyłane jako tekst jawny.

7.4 Uwierzytelnienie Digest.

Ten mechanizm uwierzytelniający jest lepiej zabezpieczony od poprzednika (daje on ochronę przed większością defektów uwierzytelnienia typu *Basic*), lecz nadal w konfrontacji z nowoczesnymi standardami kryptograficznymi jest stosunkowo słaby.

Bazuje na wykorzystaniu współdzielonego sekretu razem z uwierzytelnieniem metodą wyzwanie/odpowiedź (challenge / response) oraz zastosowaniem funkcji skrótu (hash function).

Przebieg realizacji omawianej usługi dla mechanizmu *Digest* jest następujący: gdy klient otrzymuje wyzwanie z odpowiedzią *401 Authentication Required*, która zawiera w nagłówku pole *WWW-Authenticate* wykorzystuje swoje hasło i część informacji przeniesionych z tego właśnie pola do wyliczenia skrótu. Jest on następnie dołączany do nagłówka w pole *Authorization* w ponownie wysylnym do serwera żądaniu. Obrazuje to następujący rysunek:



Rys. 8. Przebieg uwierzytelnienia *Digest*

Do wypełnienia wspomnianego pola nagłówka *Authorization*, *SIP Digest* wykorzystuje się po stronie klienta charakterystyczne dla niego informacje (zwykle kombinacje nazwy użytkownika i hasła) w połączeniu z parametrem *nonce* dostarczonym przez serwer. Dopiero taki klucz jest stosowany do funkcji haszującej MD5 (Message Digest 5). Algorytm wykorzystuje właśnie ten połączony ciąg znaków i „skraca” go do postaci 128-bitowej. Następnie wysyła otrzymany skrót do serwera, który zna dane użytkownika (shared secret) łączy je z parametrem *nonce* i również stosuje tą samą funkcję haszującą. Kolejnym etapem jest porównanie obu skrótów i jeśli okazują się być takie same tym samym klient zostaje uwierzytelniony.

Ważnym aspektem w odniesieniu do zagwarantowania bezpieczeństwa uwierzytelnienia mechanizmem *Digest* jest odpowiedni dobór parametru *nonce*, który charakteryzuje bieżące wyzwanie. Powinien być on unikalny dla każdej wymiany uwierzytelniającej. Jest on tworzony zwykle, w zależności od implementacji, z wykorzystaniem charakterystycznego klucza serwera, adresu IP klienta, znaku czasowy (timestamp) itp. Poprzez umiejętny dobór tego parametru można uniemożliwić przeprowadzenie udanego ataku typu powtórka.

Podobna sytuacja następuje w przypadku potrzeby uwierzytelnienia serwera na rzecz klienta, z tą różnicą, że na nie uwierzytelnione żądanie serwera klient wysyła odpowiedź: *407 Proxy Authorization Required*.

Po uwierzytelnieniu klienta kolejne odpowiedzi serwera mogą zawierać pola nagłówka *Authentication-Info*, aby zapewnić wzajemne uwierzytelnienie (Mutual Authentication). Stosuje się je po to, aby uniknąć sytuacji, w której trzeba by było przy każdym nowym żądaniu powtarzać pełny mechanizm uwierzytelnienia. Działoby się tak, ponieważ serwer przy każdej nowej wiadomości sygnalizacyjnej generuje nową wartość parametru *nonce*, a tak przy wykorzystaniu wspomnianego wyżej pola można poinformować go o zmienionej wartości potrzebnego do obliczenia skrótu parametru.

SIP Digest może oferować prymitywną formę wsparcia usługi integralności danych poprzez dodanie do informacji poddanych funkcji skrótu ciała wiadomości lub określonych nagłówków. Jednak w takiej formie protokół SIP nie wspiera uwierzytelnienia odpowiedzi oraz nie przeprowadza się uwierzytelniania w relacji *proxy-proxy*. Obie przedstawione wersje mechanizmu *SIP Digest* zapewniają integralność wiadomości wyłącznie w szczątkowej formie, co kłóci się z definicją uwierzytelnienia założoną przez mnie przy definiowaniu wzajemnych relacji pomiędzy poszczególnymi usługami (oraz samych usług). Biorąc pod uwagę to i inne „defekty” obu opisanych dotychczas mechanizmów należy wysnuć wniosek, że jakość uwierzytelnienia przez nie gwarantowana jest bardzo niska.

7.5 Uwierzytelnienie typu Hop-by-Hop

Protokoły gwarantujące uwierzytelnienie tego typu to: protokół warstwy transportowej TLS (Transport Layer Security) oraz warstwy sieciowej IPsec (Internet Protocol Security). Nie zostały one obowiązkowo narzucone w RFC 2543 – zakłada się tam tylko, że jeśli trzeba będzie zapewniać uwierzytelnienie Hop-by-Hop to można użyć jednego z tych dwóch protokołów. Należy również przypomnieć, że oba protokoły realizują pełną usługę integralności wiadomości, czego niestety nie gwarantują mechanizmy uwierzytelnienia typu End-to-End w tym zaleceniu, co dowiodłem powyżej.

7.6 Poufność

Zapewnienie poufności wiadomości gwarantuje, że tylko upoważnione do tego osoby są w stanie odczytać jej zawartość. Realizację tej usługi osiąga się w SIP poprzez zastosowanie mechanizmu szyfrowania. Nie ma narzuconych systemów kryptograficznych dla realizacji tej usługi, a zatem potencjalnie istnieje łatwa rozszerzalność praktycznie zaś brak narzuconego rozwiązania zwykle owocuje brakami implementacyjnymi.

Wiadomości SIP (żądania i odpowiedzi) mogą zawierać ważne dane dotyczące połączenia, jak również w samym ciele wiadomości znajdować mogą się klucze użyte do szyfrowania danej sesji. Zapewnienie prywatności połączenia poprzez szyfrowanie oznacza zabezpieczenie przed dostępem osób trzecich do informacji o agencie użytkownika tj. niemożliwość pozyskania adresu IP czy numeru portu. Dodatkowo maskowane są parametry zestawianego połączenia. SIP obsługuje trzy uzupełniające się formy szyfrowania:

- **End-to-End** – gdzie szyfrowaniu podlega całe ciało wiadomości oraz istotne z punktu widzenia bezpieczeństwa pola nagłówka. Żądania i odpowiedzi nie mogą być w ten sposób szyfrowane w całości, ponieważ wymagana jest dostępność niektórych pól np. *To* i *Via* dla serwerów proxy po to, aby wiadomość mogła zostać poprowadzona właściwą drogą.

W RFC 2543 zaleca się wykorzystywać przy szyfrowaniu dwa pola nagłówka: *Encryption* oraz *Response-Key* – jeden do wskazania, że szyfrowanie zostało użyte w wiadomości, a drugie zawiera klucz odpowiedzi. Realizacja tego szyfrowania oparta jest na kluczach współdzielonych przez obu agentów uczestniczących w wypełnieniu żądania. Zwykle wiadomość jest zaszyfrowana z użyciem klucza publicznego odbiorcy, żeby tylko on był w stanie ją przeczytać. Wszystkie implementacje bazujące na tym zaleceniu powinny obsługiwać szyfrowanie na bazie PGP (Pretty Good Privacy).

- **Hop-by-Hop** – zapobiegająca możliwości wytropienia, kto dzwoni, do kogo. Szyfrowana jest w tym przypadku cała wiadomość SIP. Możliwe jest również zaszyfrowanie wiadomości, na której uprzednio została przeprowadzona metoda *End-to-End*. Jednak w takiej sytuacji serwery proxy nadal muszą być w stanie zidentyfikować rozmawiających, a zatem identyfikacja stron jest łatwa do ustalenia np. poprzez przeprowadzenie analizy ruchu w sieci (co stanowi to o ograniczoności tego rodzaju zabezpieczenia). Realizacja takiego typu szyfrowania jest możliwa w warstwie sieciowej lub transportowej modelu TCP/IP. Do tego celu wykorzystuje się jeden z dwóch mechanizmów: IPsec lub TLS (podobnie jak w przypadku uwierzytelnienia). Inną odmianą tej wersji szyfrowania jest szyfrowanie *Hop-by-Hop* pola *Via* – w celu ukrycia trasy, którą podjęła dana wiadomość. Pole *Via* jak już wcześniej napisałem jest używane po to, aby odpowiedź mogła zostać wysłana tą samą drogą, którą przyszło żądanie oraz w celu likwidacji nieskończonych pętli żądań. Szyfrowanie tego pola jest jednak możliwe bez utraty jego funkcjonalności poprzez odpowiednią współpracę serwerów proxy i wykorzystanie odpowiedniego pola nagłówka (*Hide: hop*).

Wobec powyższych faktów należy jasno stwierdzić, że szyfrowania typu *Hop-by-Hop* nie są wspierane bezpośrednio przez protokół SIP – wykorzystują do tego wspomniane protokoły warstw niższych.

W przypadku wykorzystania szyfrowania wszystkie pola nagłówka, które pozostaną niezasyfrowane muszą poprzedzać te zaszyfrowane. Obowiązkowe w związku z tym jest wystąpienie dodatkowo dwóch pól:

- Pola *Encryption*, które wskazuje na wykorzystanie w danej wiadomości mechanizmów szyfrowania,
- Pola *Content-Length*, wskazującego długość zaszyfrowanego ciała wiadomości.

Każde pole nagłówka, które zostało zaszyfrowane w przychodzącym żądaniu powinno mieć taki sam status w odpowiedzi.

Dodatkowo:

Istnieje możliwość **szyfrowania przez serwery proxy** żądań SIP (jedyne przypadki) - gdy system końcowy sam nie może wykonać tej operacji.

8 Architektura bezpieczeństwa SIP w zaleceniu RFC 3261

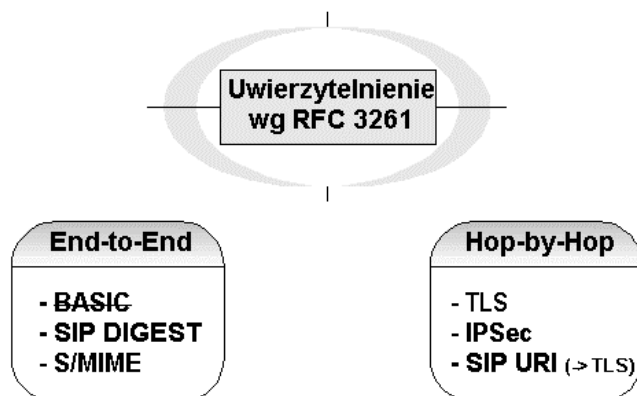
Nowe zalecenie konstytuujące drugą wersję protokołu SIP zostało opublikowane w czerwcu 2002 roku. Uwzględnia ono aspekty zapewnienia bezpieczeństwa w dużo szerszym zakresie niż pierwsza wersja zawarta w RFC 2543. Wymienia się tam potencjalne ataki na protokół SIP, a następnie charakteryzuje mechanizmy, które pozwalają im zapobiegać. Zweryfikowane zostały również stare mechanizmy i dodane nowe. Konwencja tworzenia protokołu SIP polegała właśnie na tym, aby wykorzystywać możliwie jak najwięcej istniejących już i sprawdzonych komponentów (mechanizmów) jest tak zarówno w nowym jak i starym zaleceniu. SIP wykorzystuje mechanizmy zaprojektowane oczywiście przez IETF.

Architekturę bezpieczeństwa tego zalecenia będziemy rozpatrywać zgodnie z przyjętym kryterium dla umiejętności zagwarantowania usług uwierzytelnienia oraz poufności (podobnie jak w przypadku wersji pierwszej tego protokołu) dla dwóch grup mechanizmów: End-to-End oraz Hop-by-Hop.

Na początek należy wspomnieć, iż w zaleceniu RFC 3261 mechanizmy bezpieczeństwa typu Hop-by-Hop zostały zapewnione (dla obu usług) poprzez warstwy niższe – czyli transportową (TLS) oraz sieciową (IPSec) modelu TCP/IP.

8.1 Uwierzytelnienie typu End-to-End

W stosunku do poprzedniej wersji protokołu SIP zrezygnowano ze stosowania uwierzytelnienia typu *Basic*, natomiast konieczne stało się implementowanie *SIP Digest*, które zostało nieznacznie zmodyfikowane. Dodatkowo dodano możliwość skorzystania z nowego rozwiązania zwanego *S/MIME* (*Secure / Multipurpose Internet Mail Extension*). Zatem realizacja usług uwierzytelnienia w tym zaleceniu przedstawia się następująco:



Rys. 9. Realizacja usługi uwierzytelnienia wg RFC 3261

Przejdę teraz do opisu i analizy nowych mechanizmów tutaj zastosowanych.

8.2 *Uwierzytelnienie wzajemne z wykorzystaniem S/MIME*

S/MIME jest to protokół zapewniający bezpieczeństwo protokołowi MIME, który jest z kolei głównym standardem wykorzystywanym do (de)kodowania plików o różnych formatach oraz znaków narodowych.

Wiadomości SIP zawierają ciała MIME, a ten standard definiuje protokół S/MIME do zapewnienia usługi uwierzytelnienia i poufności.

S/MIME oferuje możliwość podpisywania oraz szyfrowania jednostek MIME (czyli zapewnienia obu usług ochrony informacji i komunikacji w sieci: uwierzytelnienia oraz poufności). Dla zagwarantowania pierwszej z nich S/MIME wykorzystuje tunelowanie. Osiąga się to poprzez wykonanie pełnej lub częściowej kopii nagłówków wiadomości SIP i umieszczenie jej wraz z oryginalnym ciałem w wiadomości „wewnętrznej”. To właśnie ta „wewnętrzna” wiadomość jest enkapsulowana w jednostce MIME, która reprezentuje ciało nowej wiadomości. Może ona zostać również podpisana cyfrowo z wykorzystaniem S/MIME. W takim przypadku oryginalne nagłówki są umieszczane w wiadomości „zewnątrznej” i mogą być modyfikowane podczas transmisji. Po osiągnięciu celu odbiorca wiadomości weryfikuje dostarczony podpis cyfrowy. Jeśli podpis jest prawidłowy to dodatkowo należy dokonać porównania nagłówków wiadomości „wewnętrznej” i „zewnątrznej” oczywiście tych, które nie były wykorzystywane przez urządzenia znajdujące się na drodze komunikacyjnej, ponieważ one mogły być modyfikowane. Jest to po prostu zapewnienie podstawowej integralności wiadomości. Jeśli taka procedura przebiegnie prawidłowo oznacza to, iż wiadomość nie została zmodyfikowana w trakcie transmisji.

8.3 *Uwierzytelnienie typu Hop-by-Hop*

W przypadku tego rodzaju uwierzytelnień pozostały sprawdzone z pierwszej wersji SIP rozwiązania wykorzystujące protokoły warstw niższych modelu TCP/IP: transportowej - TLS oraz sieciowej - IP Sec. Nowością dodaną w RFC 3261 jest mechanizm o nazwie SIPS URI.

8.4 *Uwierzytelnienie SIPS URI*

Jeśli zwykły URI (Uniform Resource Identifier) postaci **sip://JKowalski@pw.edu.pl** zamienimy na **sips://JKowalski@pw.edu.pl** będzie to oznaczać fakt, iż cel wyszczególniony w adresie ma zostać osiągnięty w sposób bezpieczny, co oznacza, iż wszystkie urządzenia oraz cała droga powinna być zabezpieczone z wykorzystaniem protokołu TLS. Jeśli taki warunek nie może zostać spełniony to nie dochodzi do nawiązania połączenia. Innymi słowy, jeśli jako mechanizm gwarantujący uwierzytelnienie został wybrany SIPS URI oznacza to, że musi zostać zaimplementowany w takiej sieci protokół TLS, gdyż jest on niezbędny do prawidłowego jego funkcjonowania.

W przypadku nie stosowania mechanizmu SIPS URI, jeśli usługa poufności jest gwarantowana w „zwykły” sposób z wykorzystaniem TLS oznacza to jedynie, że odcinek pomiędzy agentem użytkownika wysyłającym żądanie, a pierwszym serwerem jest zabezpieczony za pomocą tego protokołu.

8.5 *Poufność*

W przypadku gwarantowania tej usługi ochrony informacji w porównaniu z pierwszą wersją protokołu SIP nastąpiła całkowita rezygnacja z szyfrowania typu End-to-End za pomocą PGP.

Została ona zastąpiona opcjonalnym wykorzystaniem do tego celu protokołu S/MIME, który już na pierwszy rzut oka wydaje się być lepszym, bardziej intuicyjnym rozwiązaniem niż PGP. Niestety nie wiadomo, czemu organizacja IETF nie zastosowała protokołu S/MIME już w pierwszej wersji SIP.

Szyfrowanie typu Hop-by-Hop – patrz początek rozdziału (dla przypomnienia: TLS oraz IPSec).

8.6 *Poufność z wykorzystaniem protokołu S/MIME*

Protokół S/MIME jest stosowany (jak wspomniano już wcześniej) do zagwarantowania bezpieczeństwa jednostkom MIME. W poprzednim punkcie opisałem już, w jaki sposób zagwarantować usługę uwierzytelnienia z wykorzystaniem tego protokołu teraz kolej na usługę poufności.

Poufność ciała wiadomości SIP jest realizowana z wykorzystaniem szyfrowania, a nagłówków z wykorzystaniem tunelowania oraz szyfrowania. Tunelowanie nagłówków wiadomości polega na ukryciu ich w „wewnętrznej” wiadomości, podczas, gdy w nagłówkach „zewnętrznych” znajduje się tylko część zawartych w nich informacji. Szyfrowanie bazuje na technice klucza publicznego i powinno wykorzystywać jako funkcji skrótu: SHA-1, a jako mechanizmu szyfrującego 3DES (inne są również akceptowane).

9 Słabe punkty w architekturze bezpieczeństwa SIP w zaleceniu RFC 2543

Bezpieczeństwo systemu jest tak mocne jak mocne jest jego najsłabsze ogniwo. Nikt nie będzie wykorzystywał protokołu SIP do realizacji usługi telefonii, jeśli nie będzie on gwarantował satysfakcjonującego poziomu bezpieczeństwa. Jasną sprawą jest fakt, że skoro powstała druga wersja protokołu SIP to pierwsza łagodnie ujmując nie spełniała oczekiwań twórców oraz użytkowników. Głównie chodzi tu właśnie o aspekty bezpieczeństwa (wykazano wiele słabych punktów i niedociągnięć SIP z zalecenia RFC 2543 rozważanego w aspekcie zapewnienia dwóch podstawowych usług ochrony informacji).

Zastosowane tu mechanizmy wywodzące się z samego protokołu SIP (czyli bez grupy mechanizmów Hop-by-Hop) są niewystarczające, a zastosowane systemy kryptograficzne w dużej mierze przestarzałe w porównaniu z tymi uważanymi za należące do nowoczesnej kryptografii. Głównymi „grzechami” SIP w zaleceniu RFC 2543 jest:

- Zbyt duża opcjonalność w wyborze mechanizmów gwarantujących wymagane usługi oraz brak jasnego narzucenia i przypisania konkretnych mechanizmów bezpieczeństwa poszczególnym częściom drogi komunikacyjnej,
- Brak gwarancji integralności wiadomości przy uwierzytelnieniu typu End-to-End: ani *Basic* ani *Digest*, co automatycznie uniemożliwia spełnienia usługi uwierzytelnienia,
- Zastosowanie do uwierzytelnienia mechanizmów *Basic* (całkowita nie odporność na atak typu powtórka),
- Niewystarczające bezpieczeństwo z *SIP Digest* - działanie na zasadzie współdzielonego sekretu (tak samo *Basic*),
- Dla realizacji usługi poufności wykorzystanie głównie szyfrowania PGP. Obecnie nie jest to najbezpieczniejszy system kryptograficzny szczególnie w starszych wersjach głównie, dlatego iż brak mu systemu certyfikacji oraz gdyż używane długości kluczy kryptograficznych są zbyt krótkie. Kolejnym minusem jest fakt wykorzystania jako funkcji skrótu MD5 (na którą został już przeprowadzony udany atak);

10 Słabe punkty w architekturze bezpieczeństwa SIP w zaleceniu RFC 3261

Zalecenie RFC 3261 jest jeszcze zbyt nowe i dotychczas nie opublikowano udanych ataków na nie, ponieważ nie zaimplementowano dotychczas zbyt wiele komercyjnych systemów bazujących na protokole SIP w wersji drugiej. Wydaje się, że architektura bezpieczeństwa oferowana w tym zaleceniu jest poprawna, aczkolwiek nie wystrzeżono się kilku słabości (z których większość nie jest bezpośrednio związana z samym protokołem SIP). Przejdźmy jednak do ich analizy.

Przypomnijmy: w RFC 3261 nie ma uwierzytelnienia typu *Basic*, ale za to *Digest* jest obowiązkowe. Jest dodatkowo opcjonalne S/MIME realizujące uwierzytelnienie wzajemne.

W przypadku mechanizmów realizujących usługi *Hop-by-Hop* protokół TLS musi być obowiązkowo implementowany (ale działa on niestety tylko na TCP) lub IP Sec (wybór opcjonalny).

Pomimo, że mechanizmy bezpieczeństwa dostarczone w SIP w omawianym zaleceniu redukują ryzyko ataku posiada on kilka słabości. Są one następujące:

- **SIP Digest** – pozostawione odkryte niektóre nagłówki (inaczej się nie da!), które muszą być wykorzystywane przez pośrednie urządzenia. Nie ma gwarancji integralności – można jedynie dodać kilka nagłówków, które następnie zostaną użyte do obliczenia skrótu.

- **S/MIME** – brak infrastruktury wymiany kluczy publicznych - powinien być jakiś system wymiany kluczy – ten zdefiniowany w wersji drugiej SIP jest nieodporny na atak typu *man-in-the-middle* (podobnie jak w innych systemach np. w SSH). W sytuacji, gdy atakujący przechwyci pierwszą wymianę kluczy pomiędzy komunikującymi się i będzie miał szansę przechwytywania całego dialogu między stronami komunikującymi się przeprowadzony atak można będzie uznać za udany.
Druga rzecz – wykorzystanie S/MIME może owocować dużymi (w sensie objętości) wiadomościami. Kolejnym problemem jest fakt, iż jeśli na drodze wiadomości znajdzie się jakiś rzadki typ serwera sieciowego (nie typowy proxy), którego prawidłowe działanie zależy od możliwości dostępu i modyfikowaniu ciała wiadomości SIP to wtedy protokół S/MIME uniemożliwi prawidłowe funkcjonowanie takiego elementu sieciowego.
- **TLS (mechanizm nie wynikający bezpośrednio z SIP)** – nie funkcjonuje na UDP – tylko TCP – ciągle, długo trwające połączenia TLS-over-TCP oraz uwierzytelnienie tylko serwerów, do których przylegają jednostki SIP (niedogodność ta występuje również w przypadku zalecenia RFC 2543).

Dla obu wersji protokołu SIP typem ataku, przed którym nie ma całkowitej ochrony jest atak typu blokowanie działania (Denial of Service). Bez względu na rodzaj zaimplementowanych mechanizmów bezpieczeństwa zawsze możliwe jest „zalanie” serwera (głównie chodzi tu o serwery proxy) poprzez wysyłanie nadmiernej ilości zwykle niepoprawnych wiadomości, w ten sposób powodując odmowę świadczenia usług, dla których dana jednostka została stworzona. W efekcie może to doprowadzić do stanu, gdy określony ruch pomiędzy stronami komunikującymi się nie będzie mógł poprawnie osiągnąć celu.

Niestety tego typu ataku nie da się wyeliminować całkowicie, ponieważ wiązałoby się to z ograniczeniem podstawowych funkcji serwerów sieciowych, dla których zostały one stworzone (trudno sobie wyobrazić np. serwer proxy nie obsługujący zgłoszeń połączeń).

Jedynym rozwiązaniem, które może w sposób satysfakcjonujący ograniczyć prawdopodobieństwo wystąpienia takiego ataku jest przeprowadzanie wzajemnego uwierzytelnienia serwerów proxy poprzez wykorzystanie protokołu TLS, który powinien być obowiązkowo uwzględniany przy aplikacjach bazujących na drugiej wersji protokołu SIP.

11 Przeprowadzone doświadczenia praktyczne

W ramach potwierdzenia wyników przeprowadzonych analiz mechanizmów bezpieczeństwa dla protokołu SIP w dwóch wersjach przeprowadzono badania praktyczne. Ich przebieg, wykorzystane aplikacje oraz testy zostaną zaprezentowane poniżej.

11.1 Przebieg badań

Całość przeprowadzonych działań praktycznych polegała na wykonaniu szeregu testów na aplikacjach będących implementacjami Agentów Użytkownika SIP. Pierwotnym celem było zbadanie odporności na różnego rodzaju próby ataków mechanizmów bezpieczeństwa w nich zawartych. W zamierzeniu miała być to po prostu symulacja potencjalnych, celowych prób działań atakującego i na tej podstawie ocena umiejętności radzenia sobie w przypadku ich wystąpienia.

Niestety ilość i różnorodność ataków na protokół SIP jest znaczna i testując samą aplikację Agent Użytkownika SIP nie odda się całego spektrum potencjalnych zagrożeń. Jednakże uznając, że w praktyce użytkownik nie ma zbyt wielkiego wpływu na bezpieczeństwo innych komponentów architektury funkcjonalnej SIP tylko na SIP UA, właśnie na testowanie tego elementu funkcjonalnego położyłem nacisk.

Jeśli chodzi o same testy zostały dobrane odpowiednio tak, aby oddawały jak najwięcej klas zagrożeń – kładąc szczególny nacisk na ataki najłatwiejsze do wykonania a tym samym najbardziej prawdopodobne. Wychodząc z założenia, iż intruz ma możliwość (co najmniej) przeprowadzenia podsłuchania pakietów (co w obecnych czasach nie sprawia większych trudności) i w ten sposób generowania fałszywych wiadomości.

Niestety po zapoznaniu się z dostępnymi mi aplikacjami będącymi implementacjami Agent Użytkownika pierwotny cel testowania musiał zostać zmodyfikowany. Stało się tak, ponieważ:

- Żadna z testowanych przeze mnie aplikacji (prócz jednej) nie miała zaimplementowanego, choć najprostszego mechanizmu bezpieczeństwa.

- Wszystkie aplikacje bazowały wyłącznie na starym zaleceniu SIP (RFC 2543).

W związku z faktami, które przytoczyłem powyżej testowanie Agentów Użytkownika ograniczyło się do osiągnięcia dwóch celów:

- Zbadania zgodności implementacji wybranych aplikacji z zaleceniem, na którym bazuje (w przypadku testowanych przeze mnie aplikacji jest to RFC 2543).
- Wykazanie wyższości programu, w którym został zaimplementowany, choć prosty mechanizm bezpieczeństwa opisany w zaleceniu nad aplikacją nie posiadającą żadnych wspomnianych mechanizmów (zatem jedynym mechanizmem, którego skuteczność w praktyce przetestowałem był SIP Digest).

Postacie wykorzystanych w doświadczeniach testów zostały umieszczone w plikach tekstowych tak, aby umożliwić ich łatwe tworzenie, dodawanie oraz modyfikację. Ich postać zostanie przedstawiona i opisana w dalszej części pracy oraz w załączniku A.

Do przeprowadzenia doświadczeń wykorzystana została autorska, pomocnicza aplikacja S2C (SIP Security Call Checker), która pozwoli na uwidocznienie niektórych słabości w architekturze bezpieczeństwa badanych aplikacji oraz zbadanie zgodności implementacji tychże z zaleceniem. Jej głównym zadaniem jest wysłanie określonego testu do wskazanego Agentu Użytkownika SIP, a następnie zbadanie jego reakcji.

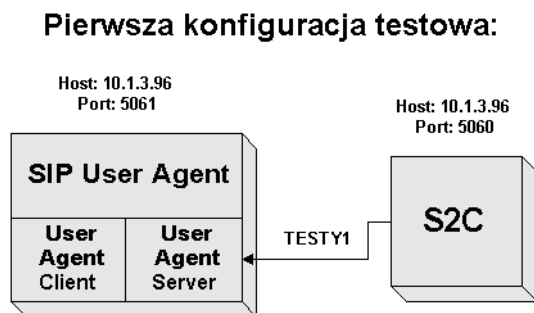
12 Opis doświadczeń i wykorzystanych testów

Postać testów została napisana specjalnie w taki sposób, aby oddać specyfikę wymiany wiadomości sygnalizacyjnych pomiędzy dwoma Agentami Użytkownika SIP. Przy ich opracowaniu bazowałem na testach dla SIP, które zostały stworzone przez twórców SIP'a: Neila Deasona, Andersa Kristensena, Jonathana Rosenberga oraz Henninga Schulzrinna.

Całość testów została przeprowadzona na jednym komputerze z zainstalowanym systemem operacyjnym Windows XP oraz kartą sieciową (jednak ma przeciwskazań ani żadnych problemów, aby przeprowadzić je w sieci np. LAN). Zainstalowałem na nim wszystkie potrzebne do doświadczeń aplikacje do testowania (przedstawione w następnej części pracy) jak i swoje narzędzie testujące (S2C). W celu symulacji ataków na sygnalizację SIP zastosowałem odpowiednie konfiguracje doświadczeń, które przedstawiam poniżej.

12.1 Konfiguracja doświadczeń

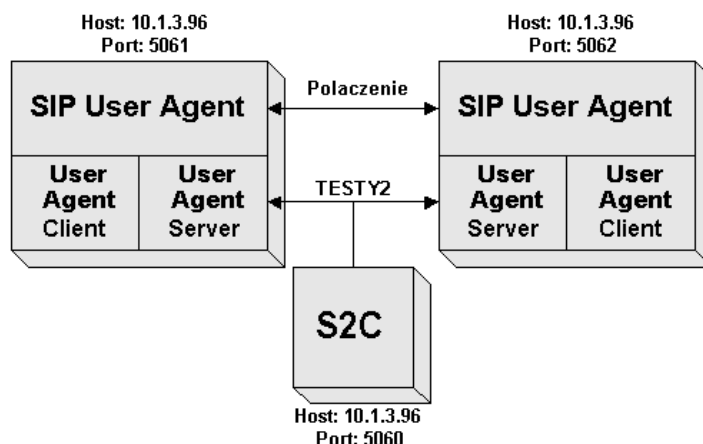
- a) **Konfiguracja1** - w tej konfiguracji badane było zachowanie testowanej aplikacji, do której przesyła się odpowiednio sporządzone wiadomości-testy. Ma to na celu zarówno sprawdzenie zgodności zachowań z zaleceniem RFC 2543 oraz symulację ataku, w którym intruz może się podszyc pod czyjąś tożsamość.



Rys. 10. Konfiguracja testowa pierwsza

- b) **Konfiguracja2** - w tym przypadku nawiązywane było połączenie pomiędzy dwoma wybranymi aplikacjami Agentu Użytkownika SIP, a następnie za pomocą aplikacji S2C oraz poprzez określone testy próbowano wpłynąć w negatywny sposób na wymianę wiadomości sygnalizacyjnych/pakietów RTP (wiadomości-testy były raz wysyłane do jednej raz do drugiej testowanej aplikacji). Była to próba symulacji ataku aktywnego, w którym intruz nie podsłuchuje komunikujących się między sobą agentów, jednak może wysyłać odpowiednio zaadresowane i sporządzone wiadomości w celu uniemożliwienia nawiązania lub przerwania trwającego połączenia.

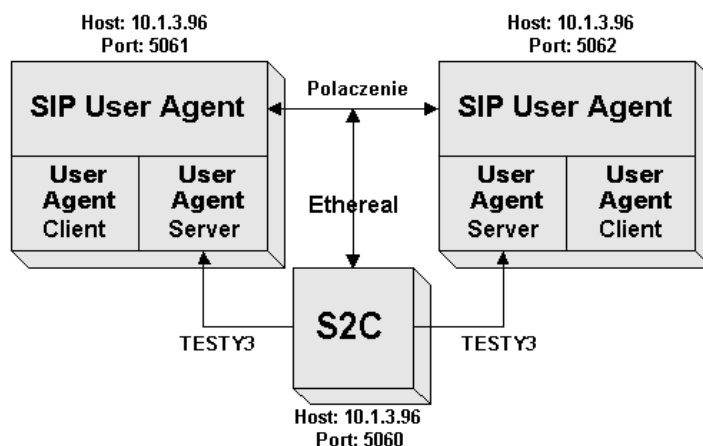
Druqa konfiguracja testowa:



Rys. 11. Konfiguracja testowa druga

e) Konfiguracja3 – Sytuacja tutaj jest podobna do przypadku opisanego w poprzedniej konfiguracji. Badana była tu również odporność połączenia na przeprowadzane na nim prób ataków (wiadomości-testy były raz wysyłane do jednej raz do drugiej testowanej aplikacji). Główną różnicą pomiędzy doświadczeniami w punktach b i c było założenie, iż atakujący ma możliwość podsłuchiwania wiadomości sygnalizacyjnych (ja do tego celu wykorzystałem popularną aplikację *Ethereal*) oraz odpowiednią ich modyfikację – głównie chodziło w tym przypadku o pokazanie jak łatwo wpłynąć na połączenie w przypadku nie wykorzystywania żadnych mechanizmów bezpieczeństwa (poprzez znajomość nagłówka: *Call-Id*).

Trzecia konfiguracja testowa:



Rys. 12. Konfiguracja testowa trzecia

12.2 Opis treści testów

Dla każdej konfiguracji przeprowadzane testy nie były identyczne. Odpowiedni ich dobór został dokonany w zależności od rodzaju wybranej konfiguracji.

Dodatkowo należy w tym miejscu koniecznie jeszcze raz pokreślić, iż postać testów dla podanych powyżej konfiguracji nie skupia się na badaniu opisanych wcześniej mechanizmów bezpieczeństwa (jest jeden wyjątek w przypadku aplikacji firmy Helmsman), lecz głównie na:

- Próbach przerwania lub zakłócenia w jak najprostszy sposób trwającego lub zestawianego połączenia. Celem takiego podejścia jest wykazanie potrzeby implementacji opisanych w zaleceniach mechanizmów bezpieczeń-

stwa – nawet tych prostych, gdyż finalnie mogą one utrudnić, choć trochę możliwość ingerencji w komunikację intruza.

- Analizie zgodności implementacji Agentu Użytkownika z zaleceniem na tych obszarach, które mogą mieć krytyczne znaczenie dla bezpieczeństwa zarówno działania samej aplikacji jak i możliwej wymiany wiadomości sygnalizacyjnych.

12.3 Grupa testów do Konfiguracji1

Głównym celem tej grupy testów jest zbadanie zgodności zachowania badanej aplikacji (pod kątem odbioru celowo źle sformowanych wiadomości sygnalizacyjnych) z zaleceniem, na którym bazuje (RFC 2543). Testy te mają wykazać ewentualne braki implementacyjne mogące wspomagać działania potencjalnego intruza, który znając takie „dziury” może utrudniać prawidłowe funkcjonowanie Agentu Użytkownika, a w rezultacie obniżyć *QoS* (*Quality of Service*) całej usługi. Wszystkie treści wykorzystanych w tej konfiguracji testów zostały umieszczone w Dodatku A.

12.4 Grupa testów do Konfiguracji2

Celem przeprowadzania tej grupy testów jest chęć sprawdzenia szans intruza, który będzie próbował przerwać lub zakłócić istniejące połączenie. Zakładam, iż będzie on działał „na ślepo” tzn. zna tylko adres SIP jednej z komunikujących się stron. Wszystkie treści wykorzystanych w tej konfiguracji testów zostały umieszczone w Dodatku B.

12.5 Grupa testów do Konfiguracji3

Celem przeprowadzania tej grupy testów jest chęć sprawdzenia szans intruza, który będzie próbował przerwać lub zakłócić istniejące połączenie. Zakładam, że atakujący za pomocą aplikacji podsłuchującej pakiety jest w stanie odkryć zawartość przesyłanych wiadomości. Wszystkie treści wykorzystanych w tej konfiguracji testów zostały umieszczone w Dodatku B (są to prawie te same testy, co w przypadku Konfiguracji2).

12.6 Testy mechanizmu SIP Digest

Do przeprowadzenia doświadczeń w tym punkcie wykorzystam konfigurację trzecią (tam gdzie wykorzystuję aplikację *Ethereal*) oraz odpowiednio ułożone testy, które będą odpowiadały sytuacji:

- a) Podobnej jak w Konfiguracji2, gdy intruz przeprowadza ataki „na ślepo” – znając tylko adres SIP Agentu Użytkownika.
- b) Podobnej jak w Konfiguracji3, gdy atakujący za pomocą aplikacji podsłuchującej pakiety jest w stanie odkryć zawartość przesyłanych wiadomości.

Zastosowane testy:

- a) Dowolne wybrane z grupy testów dla Konfiguracji1 oraz próba nawiązania prawidłowego połączenia w czyimś imieniu,
- b) Próba podszycia się pod innego użytkownika z wykorzystaniem informacji uzyskanych przy pomocy aplikacji *Ethereal* – zastosowanie poprawnie sformułowanej wiadomości ze zmodyfikowanymi polami pobranymi z przechwyconych wiadomości.

Zawartość testu wykorzystana do poprawnego nawiązania połączenia:

```
INVITE sip:woytek@10.1.3.96:5061 SIP/2.0
Via: SIP/2.0/UDP 10.1.3.96
CSeq: 0 INVITE
Call-ID: 98asdh@10.1.1.2
From: sip:xxx@10.1.3.96:5060
To: sip:xxx@10.1.3.96:5060
Content-Type: application/sdp
```

v=0

```
o=mhandley 29739 7272939 IN IP4 126.5.4.3
c=IN IP4 135.180.130.88
m=audio 49210 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC/8000
```

13 Testowane aplikacje SIP UA

W Internecie dostępnych było około dziesięciu darmowych aplikacji będących implementacjami Agenta Użytkownika SIP. Przy wyborze programów do testowania kierowałem się głównie ich dostępnością oraz tym by każdy z nich był firmowany przez innych twórców (by uzyskać jak najszerszy przegląd dostępnych UA). Wszystkie dostępne mi aplikacje SIP UA zostały zaimplementowane zgodnie z RFC2543. Nie miałem możliwości przetestowania **żadnych** aplikacji bazujących na RFC3261.

Do celów badawczych pokazujących możliwości zrealizowanego przeze mnie narzędzia wybrałem następujące dostępne mi darmowe programy:

- Helmsman User Agent 3.0.6 firmy Helmsman,
- eStara SoftPHONE 3.0 – firmy eStara.
- Siemens Communication System Client v.1.0 firmy Siemens,
- Magellan 4.0 opracowany w IT PW,
- Hughes SIP User Agent (E-Z Phone) firmy Hughes Software Systems,
- Vovida SIP UA 1.0.2 - Columbia University,

Należy w tym miejscu po raz kolejny podkreślić, iż z pośród wszystkich aplikacji wymienionych powyżej tylko Helmsman User Agent 3.0.6 posiada celowo zaimplementowany jakikolwiek mechanizm bezpieczeństwa (w tym przypadku jest to możliwość uwierzytelnienia *SIP Digest*). Reszta programów została zubożona on ten krytyczny obszar stanowiący o jakości oraz funkcjonalności zarówno samego Agenta Użytkownika jak i całej usługi Voice over Internet Protocol.

14 Analiza wyników przeprowadzonych testów z grup 1,2 oraz 3

Wyboru najlepszej pod względem zapewnienia bezpieczeństwa aplikacji będącej implementacją Agenta Użytkownika SIP dokonałem po analizie ocen wszystkich testów dla poszczególnych aplikacji (ilości zaliczonych testów). W trakcie doświadczeń przeprowadziłem na każdej aplikacji 25 różnych testów.

Ocenianie testów:

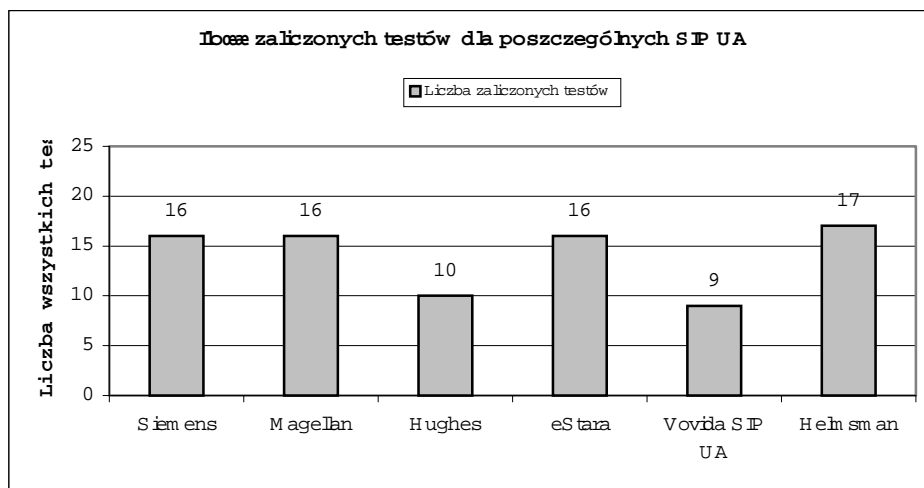
„+” – test zaliczony – zachowanie zgodne z oczekiwaniami;

„+/-” – test częściowo zaliczony (sytuacja, gdy np. nie nastąpiło przerwanie połączenia, ale nie ma sygnalizacji błędu);

„-” – test nie zaliczony.

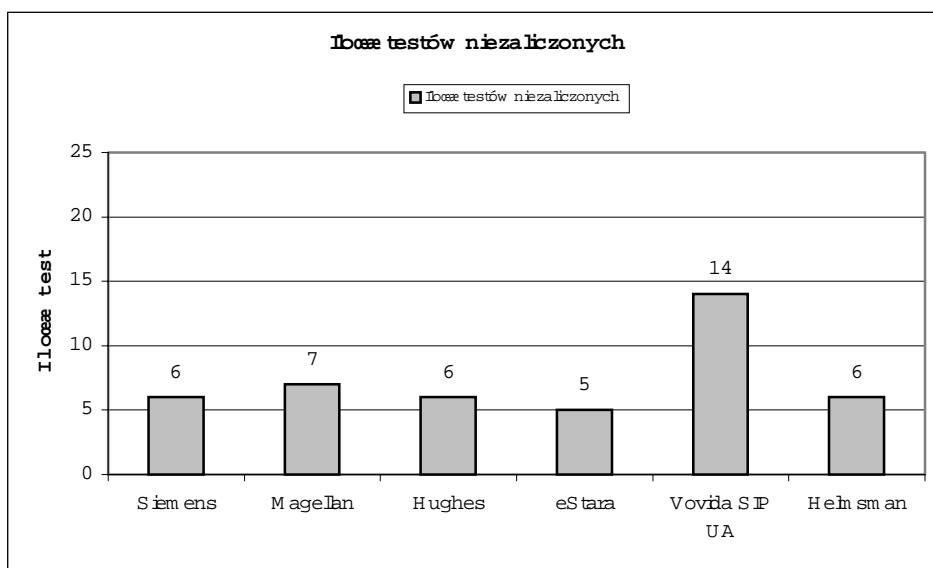
14.1 Uzyskane wyniki

Suma zaliczonych testów dla poszczególnych aplikacji przedstawia się następująco:



Rys. 13. Ilość zaliczonych testów dla poszczególnych SIP UA

Natomiast sumę testów nie zaliczonych dla poszczególnych aplikacji obrazuje wykres:



Rys. 14. Ilość testów niezaliczonych dla poszczególnych SIP UA

Analiza powyższych wykresów prowadzi do wniosku, iż spośród sześciu testowanych przeze mnie aplikacji SIP UA na wyróżnienie zasługuje aplikacja firmy Helmsman. Sprawdziła się ona zarówno pod względem zaliczonych testów, jak pod względem interfejsu i komunikacji z użytkownikiem - jest on godny uwagi – prosty, intuicyjny i funkcjonalny.

Żadna z aplikacji nie zaliczyła wszystkich testów, co dowodzi jak łatwo wcale nie dużymi środkami osiągnąć udany atak na tego rodzaju program. Wielu Agentów Użytkownika w ogóle nie reagowało na wiadomość testującą, mimo tego, że działając zgodnie z zaleceniem, na którym bazują powinny zasygnalizować wystąpienie błędu.

14.2 Wyniki i analiza testów mechanizmu SIP Digest

Jak już podkreślano jedyną aplikacją ze wszystkich mi dostępnych, która miała zaimplementowany jakiegokolwiek mechanizm bezpieczeństwa była aplikacja Helmsman User Agent 3.0.6, co uwzględniając wyniki testów przeprowadzone w poprzednim punkcie stawiają tą aplikację w bardzo korzystnym świetle i pozwalają, tym bardziej, na wyróżnienie jej spośród innych testowanych w tej pracy.

Zastosowanym w niej mechanizmem bezpieczeństwa jest *SIP Digest*. Opcja wykorzystania tego mechanizmu nie jest włączona domyślnie. Dlatego też testy przeprowadzone wcześniej na tej aplikacji nie uwzględniały takiego przypadku.

W przypadku wysłania poprawnej wiadomości bez zawartych w niej danych uwierzytelniających głównie otrzymałem odpowiedzi: 401 Unauthorized. Jest to zgodne z oczekiwaniami oraz zasadą działania SIP Digest.

Tą samą odpowiedź otrzymałem również w przypadku próby nawiązania połączenia poprawną wiadomością (poszycie się).

Aby odnaleźć wartości potrzebne do obliczenia skrótu przy uwierzytelnieniu SIP Digest dla aplikacji Helmsman UA możliwe są dwa rozwiązania:

- Odczytanie ich z odpowiedzi 401 Unauthorized, która przychodzi do nas zwrotnie po wysłaniu np. metody INVITE (nagłówek WWW-Authenticate),
- Zdobycie szukanych wartości za pomocą podsłuchania wiadomości (aplikacja *Ethereal*).

Niestety, dla atakującego, aby wymiana uwierzytelniająca zakończyła się sukcesem to skrót wyliczony z tych wartości jak również z nazwy użytkownika oraz jego hasła musi zostać wysłany w „poprawionej” wiadomości w nagłówku *Authorization*. Mimo podsłuchania tego nagłówka intruz nie ma wystarczających danych ku temu by wygenerować oczekiwany skrót.

Jedynym sposobem złamania takiego mechanizmu uwierzytelnienia jest zdobycie brakujących danych użytkownika. Dodatkowo wartość pola *nonce* jest różna dla każdej wymiany uwierzytelniającej.

Podsumowując nie jest to rozwiązanie idealne, ale w wydatny sposób poprawia bezpieczeństwo badanego Agenta Użytkownika. Widać teraz jak na dłoni, iż atakujący zmuszony jest do dużo większego wysiłku przy zastosowaniu nawet tak nieidealnego mechanizmu.

W innym przypadku naprawdę nie jest trudno wpłynąć negatywnie na samą aplikację lub połączenie – atakujący potrzebuje naprawdę minimalnej wiedzy w postaci odpowiedniej treści wiadomości oraz narzędzia umożliwiającego jej wysłanie.

15 Słabości implementacji komercyjnych SIP UA ujawnione przez organizację CERT

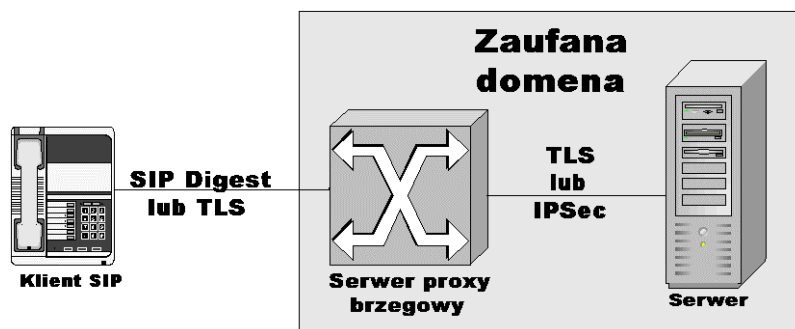
Po opracowaniu metody testowania oraz przeprowadzeniu zaprezentowanych testów bezpieczeństwa na dostępnych mi aplikacjach SIP UA w ramach pracy dyplomowej - 21 lutego 2003 roku znana organizacja CERT w artykule „*CA-2003-06 Multiple vulnerabilities in implementations of the Session Initiation Protocol (SIP)*” opublikowała niezależne wyniki specjalnie opracowanych testów PROTOS, których cel oraz natura były podobne do przeprowadzonych przez autora. Jako, że testy odbywały się były w laboratoriach ich zasięg oraz możliwości były większe (m.in. testowane również inne elementy architektury funkcjonalnej SIP tj. serwery proxy, redirect oraz registrar). Przeprowadzone tam doświadczenia wykazały głównie podobne błędy implementacyjne w testowanych SIP UA opartym na zaleceniu RFC 3261. Stąd należy wnioskować, iż nawet najbezpieczniejszy protokół, jeśli zostanie źle zaimplementowany nie gwarantuje żadnego poziomu bezpieczeństwa. Powtórzyła się, zatem sytuacja, która została opisana w tym artykule.

Należy, więc przed zakupem komercyjnego oprogramowania dla VoIP bazującym na protokole SIP dowiedzieć się dokładnie, czy i jakie mechanizmy bezpieczeństwa zostały tam zaimplementowane. Nie dopilnowanie tego może mieć krytyczne znaczenie dla bezpieczeństwa całej komunikacji przeprowadzanej za pomocą takiej aplikacji.

16 Przykład bezpiecznej konfiguracji usługi VoIP bazującej na SIP

Przy projektowaniu bezpiecznej konfiguracji dla usługi Voice over Internet Protocol wykorzystującej jako protokół sygnalizacyjny protokół SIP należy umiejętnie dobrać dostępne mechanizmy bezpieczeństwa oferowane przez ten protokół, jak również niezbędne jest wykorzystanie z mechanizmów oferowanych przez warstwy niższe (patrz opis teoretyczny).

Przykładem bezpiecznej konfiguracji zgodnej z zaleceniem RFC3261 może być system przedstawiony na rysunku poniżej.



Rys. 15. Przykład bezpiecznej konfiguracji usługi VoIP opartej na SIP z zalecenia RFC3261

Mechanizm wykorzystany do komunikacji klient SIP - serwer proxy: **TLS** lub **SIP Digest**.

Mechanizm wykorzystany do komunikacji brzegowy serwer proxy – serwer docelowy: **TLS** lub **IPsec**.

Niezbędne założenia: brzegowy serwer proxy „dopuszcza” tylko żądania od uwierzytelnionych klientów (firewall). Serwer docelowy akceptuje tylko ruch przenoszony przez brzegowy serwer proxy.

Niepodważalnymi **zaletami** przedstawionej w tym punkcie konfiguracji jest (oprócz bezpiecznej wymiany wiadomości sygnalizacyjnych):

- Umożliwienie przeprowadzenia brzegowemu serwerowi proxy uwierzytelnienia klienta typu Hop-by-Hop,
- Oddzielenie logiki realizowania usługi uwierzytelnienia od logiki danej usługi.

Wadą takiego rozwiązania może okazać się fakt, iż wszystkie elementy usługowe muszą ufać serwerowi proxy oraz to, że serwer docelowy może nie znać prawdziwej tożsamości klienta.

17 Podsumowanie i wnioski

W niniejszym artykule w części teoretycznej zostały zebrane, zanalizowane oraz ocenione mechanizmy bezpieczeństwa tworzące architekturę bezpieczeństwa SIP dla dwóch zaleceń: RFC 2543 oraz RFC 3261. Wykazano potencjalne zagrożenia, które mogą wystąpić dla SIP w wersji pierwszej (RFC 2543). Pokazano również jak rozwiązano te problemy zastępując pierwszą wersję SIP drugą (RFC 3261). Głównym wnioskiem z tej części jest możliwość utworzenia potencjalnie bezpiecznej architektury SIP – nowe zalecenie określa mechanizmy, które powinny umożliwić wydajną i bezpieczną wymianę wiadomości sygnalizacyjnych. Wszystko pozostaje teraz w rękach implementujących, ponieważ nawet najlepsze plany są niczym w przypadku, gdy nie zostaną one poprawnie zrealizowane. Przykład takiej architektury został podany w punkcie 16.

Sprawdzenie za pomocą odpowiednio dobranych testów-wiadomości sześciu dostępnych mi, darmowych Agentów Użytkownika SIP bazujących na RFC 2543 nie dostarczyło niestety optymistycznych wyników. Z sześciu aplikacji tylko jedna (Helmsman User Agent) miała zaimplementowany jakikolwiek mechanizm bezpieczeństwa (SIP Digest). To wystarczyło, aby odpowiednio utrudnić „domowy” atak na sygnalizację SIP. Reszta aplikacji niestety nie została zrealizowana w sposób należyty.

Część winy za taki stan rzeczy ponoszą sami twórcy zalecenia RFC 2543, gdyż mechanizm SIP Digest (podobnie jak i Basic) nie jest tam narzucony (wymagany) – jest to tylko opcja. Drugą przyczyną takiego stanu rzeczy jest fakt, iż większą część z tych aplikacji stanowią wersje testowe programów lub takie, które mają zachęcić do kupna jego pełnej wersji. Możliwe, że kupując pełną wersję produktu otrzymuje się program, razem z zaimplementowanymi mechanizmami bezpieczeństwa należy, więc to bezwzględnie sprawdzić przed zakupem.

Jak pokazałem w swoich doświadczeniach istnieje wiele prostych metod, za pomocą których można zakłócić wymianę sygnalizacji, przerwać połączenie lub wpłynąć negatywnie na samą aplikację. W dzisiejszym świecie nie można zakładać, iż brak jest osób, które chciałyby ingerować, z różnych względów, w bezpieczeństwo usługi telefonii tzn. wykorzystywać znajdujące się w nim luki. Nie zdawanie sobie sprawy z tego faktu jest raczej zbytnią łatwowiernością. Wystarczy spojrzeć np. na ilość włamań do systemów komputerowych w sieci Internet.

Dowodzi to konieczności uwzględnienia w trakcie implementacji Agenta Użytkownika odpowiednich (nawet najprostszyc) mechanizmów gwarantujących bezpieczeństwo. Dlatego też m.in. w drugiej wersji protokołu SIP zapisano obowiązek realizacji mechanizmu SIP Digest.

Jeśli chodzi o bezpieczeństwo sygnalizacji systemu VoIP bazującego na protokole SIP to z pewnością druga jego wersja (bazująca na RFC 3261) gwarantuje więcej trafnych i nowoczesnych mechanizmów kryptograficznych limitujących ilość przeprowadzenia udanych ataków na taki system. Nie jest to dziwne, gdyż właśnie jedną z głównych przesłanek jej stworzenia były luki bezpieczeństwa wersji pierwszej.

Z kolei, aby prawidłowo zabezpieczyć wymianę wiadomości sygnalizacyjnych z wykorzystaniem protokołu SIP należy umiejętnie dobierać dostępne mechanizmy bezpieczeństwa. A uściślając należy zdefiniować jak jednostki funkcjonalne SIP mogą dokonywać wyboru pomiędzy odpowiednimi mechanizmami podczas komunikacji tak, aby być w stanie zagwarantować podstawowe usługi ochrony informacji i komunikacji zdefiniowane w przyjętym kryterium.

Architektura bezpieczeństwa protokołu SIP przedstawiona w RFC 3261 powinna gwarantować wystarczający poziom bezpieczeństwa tzn. taki, który zminimalizuje prawdopodobieństwo przeprowadzenia udanego ataku na protokół sygnalizacyjny SIP. Na razie jednak zalecenie to jest zbyt nowe i jeszcze nie ma żadnych opublikowanych możliwych ataków na mechanizmy bezpieczeństwa tam opisane. Ale powód braku potencjalnych ataków może być również inny – niewykluczone, że doświadczenia zebrane przy tworzeniu SIP w wersji pierwszej jak i testowaniu pierwszych aplikacji SIP zaowocowały właściwym doбором mechanizmów bezpieczeństwa dla drugiej wersji tegoż protokołu.

Jeśli zatem usługa VoIP oparta na protokole sygnalizacyjnym SIP ma stać się powszechna to musi oferować, jak już wspominałem, podobny poziom jakości usług jak w przypadku telefonii klasycznej, a dodatkowo jeszcze zachęcić użytkowników czymś więcej. Nikt, sam z siebie, nie będzie inwestował swoich środków po to by w rezultacie otrzymać usługę o gorszym standardzie. Biorąc pod uwagę dodatkowo, że informacja w obecnych czasach staje się towarem i ma swoją cenę, więc zapewnienie jej bezpiecznego przesyłania wpływa w bezpośredni sposób na wzrost atrakcyjności danej usługi.

Reasumując zapewnienie bezpieczeństwa wymiany wiadomości sygnalizacyjnych w SIP może być istotnym czynnikiem wpływającym na jego popularność jako protokołu dla VoIP.

Dodatek A

Testy Konfiguracji 1:

TEST 1_1

a) Bada odporność SIP UA na nieprawidłową postać pól: *Request URI, To, From* oraz *Contact*.

Spodziewana reakcja: wystąpienie błędu Not Found.

```
INVITE imie:Jan_Kowalski SIP/2.0
To: isbn:2983792873
From: http://www.elka.pw.edu.pl
Call-ID: 0ha0isndaksdj@10.1.2.3
Cseq: 8 INVITE
Via: SIP/2.0/UDP 135.180.130.133
Content-Type: application/sdp
Contact: Jan Stanisław Malinowski <urn:ipaddr:128.0.3.3>
```

```
v=0
o=mhandley 29739 7272939 IN IP4 126.5.4.3
c=IN IP4 135.180.130.88
m=audio 49210 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC/8000
```

b) Badana jest reakcja SIP UA na brak w metodzie INVITE pól: *Call-ID, To* oraz *From*.

Spodziewana reakcja: sygnalizacja błędu żądania

```
INVITE sip:woytek@10.1.3.96:5062 SIP/2.0
Cseq: 0 INVITE
Via: SIP/2.0/UDP 135.180.130.133
Content-Type: application/sdp
```

```
v=0
o=mhandley 29739 7272939 IN IP4 126.5.4.3
c=IN IP4 135.180.130.88
m=audio 49210 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC/8000
```

c) Badana jest reakcja SIP UA na podwójne wystąpienie pól: *Call-ID* oraz *From* w metodzie INVITE.

Spodziewana reakcja: sygnalizacja błędu

```
INVITE sip:woytek@10.1.3.96:5062 SIP/2.0
Via: SIP/2.0/UDP 135.180.130.133
Cseq: 0 INVITE
Call-ID: 98asdh@10.1.1.1
Call-ID: 98asdh@10.1.1.2
From: sip:xxx@10.1.3.96:5060
From: sip:xxx1@10.1.3.96:5060
To: sip:xxx@10.1.3.96:5060
Content-Type: application/sdp
```

```
v=0
o=mhandley 29739 7272939 IN IP4 126.5.4.3
c=IN IP4 135.180.130.88
```

m=audio 49210 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC/8000

TEST 1_2

Badane jest zachowanie SIP UA w sytuacji, gdy Agent inicjujący połączenie wymaga obsługi nowych właściwości.

Spodziewana reakcja: wystąpienie odpowiedzi 420 Bad Extension

```
INVITE sip:woytek@10.1.3.96:5061 SIP/2.0
To: sip:xxx@10.1.3.96:5060
From: sip:woytek@10.1.3.96:5060
Call-ID: 0ha0isndaksdj@10.1.1.1
Require: newfeature1, newfeature2
Cseq: 8 INVITE
Via: SIP/2.0/UDP 10.1.3.96
```

TEST 1_3

Badane jest zachowanie SIP UA w sytuacji, gdy metoda INVITE zawiera ciało, które nie jest opisane przy wykorzystaniu protokołu SDP.

Spodziewana reakcja: sygnalizacja wystąpienia błędu

```
INVITE sip:woytek@10.1.3.96:5062 SIP/2.0
To: sip:xxx@10.1.3.96:5060
From: sip:xxx@10.1.3.96:5060
Call-ID: 0ha0isndaksdj@10.0.0.1
Cseq: 8 INVITE
Via: SIP/2.0/UDP 135.180.130.133
Content-Type: application/newformat
```

```
<audio>
<pcmu port="443"/>
</audio>
```

TEST 1_4

Badane jest zachowanie SIP UA w przypadku wystąpienia nieznannej metody.

Spodziewana reakcja: sygnalizacja wystąpienia błędu oraz wypisanie dostępnych metod.

```
NEWMETHOD sip:woytek@10.1.3.96:5062 SIP/2.0
To: sip:xxx@10.1.3.96:5060
From: sip:xxx@10.1.3.96:5060
Call-ID: 0ha0isndaksdj@10.0.0.1
Cseq: 8 NEWMETHOD
Via: SIP/2.0/UDP 135.180.130.133
Content-Type: application/sdp
```

v=0

```
o=mhandley 29739 7272939 IN IP4 126.5.4.3
c=IN IP4 135.180.130.88
m=audio 49210 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC/8000
```

TEST 1_5

Badane jest reakcja SIP UA w gdy wiadomość zawiera wykraczającą poza zakres wartość pola *Content-Length*.

Spodziewana reakcja: w przypadku użycia protokołu UDP sygnalizacja błędu.

```
INVITE woytek@10.1.3.96:5062 SIP/2.0
To: sip:xxx@10.1.3.96:5060
From: sip:xxx@10.1.3.96:5060
Call-ID: 0ha0isndaksdj@10.0.0.1
Cseq: 8 INVITE
Via: SIP/2.0/UDP 135.180.130.133
Content-Type: application/sdp
Content-Length: 9999
```

```
v=0
o=mhandley 29739 7272939 IN IP4 126.5.4.3
c=IN IP4 135.180.130.88
m=audio 49210 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC/8000
```

TEST 1_6

Badane jest reakcja SIP UA w gdy wiadomość zawiera wiele pól o bardzo dużej ilości znaków.

Spodziewana reakcja: brak błędu.

```
INVITE sip:woytek@10.1.3.96:5062 SIP/2.0
To: "I have a user name of extreme proportion" <sip:user@company.com:6000;other-param=1234567890somethingelse1234567890>
From: sip:xxx@10.1.3.96:5060
Call-ID:
kl24ahsd546folnyt2vbak9sad98u23naodiunzds09a3bqw0sdfbsk34poouymnae0043nsed09mfkvc74bd0cuwnms05dknw87hjpodb76f
Cseq: 1 INVITE
My-State: sldkjflzdsfaret0803adgaasd0afds0asdaasd
Via: SIP/2.0/UDP sip33.example.com
Via: SIP/2.0/UDP sip32.example.com
Via: SIP/2.0/UDP sip31.example.com
Via: SIP/2.0/UDP sip30.example.com
Via: SIP/2.0/UDP sip29.example.com
Via: SIP/2.0/UDP sip28.example.com
Via: SIP/2.0/UDP sip27.example.com
Via: SIP/2.0/UDP sip26.example.com
Via: SIP/2.0/UDP sip25.example.com
Via: SIP/2.0/UDP sip24.example.com
Via: SIP/2.0/UDP sip23.example.com
Via: SIP/2.0/UDP sip22.example.com
Via: SIP/2.0/UDP sip21.example.com
Via: SIP/2.0/UDP sip20.example.com
Via: SIP/2.0/UDP sip19.example.com
Via: SIP/2.0/UDP sip18.example.com
```

Via: SIP/2.0/UDP sip17.example.com
Via: SIP/2.0/UDP sip16.example.com
Via: SIP/2.0/UDP sip15.example.com
Via: SIP/2.0/UDP sip14.example.com
Via: SIP/2.0/UDP sip13.example.com
Via: SIP/2.0/UDP sip12.example.com
Via: SIP/2.0/UDP sip11.example.com
Via: SIP/2.0/UDP sip10.example.com
Via: SIP/2.0/UDP sip9.example.com
Via: SIP/2.0/UDP sip8.example.com
Via: SIP/2.0/UDP sip7.example.com
Via: SIP/2.0/UDP sip6.example.com
Via: SIP/2.0/UDP sip5.example.com
Via: SIP/2.0/UDP sip4.example.com
Via: SIP/2.0/UDP sip3.example.com
Via: SIP/2.0/UDP sip2.example.com
Via: SIP/2.0/UDP sip1.example.com
Via: SIP/2.0/UDP
host.example.com;received=135.180.130.133;branch=C1C3344E2710000000E299E568E7potato10potato0potato0
Content-Type: application/sdp

v=0
o=mhandley 29739 7272939 IN IP4 126.5.4.3
s=SIP Call
c=IN IP4 135.180.130.88
t=3149328700 0
m=audio 49210 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtptime:31 LPC/8000

TEST 1_7

a) Badane jest reakcja SIP UA w gdy w wiadomości podana jest dobra wersja protokołu SIP, ale nieprawidłowo zapisana (zamiast „2.0” jest „02.00”)

Spodziewana reakcja: odrzucenie żądania odpowiedzią 400.

INVITE sip:woytek@10.1.3.96:5062 SIP/02.00
Via: SIP/2.0/UDP 10.1.3.96
From: A. Bell <sip:xxx@10.1.3.96>
To: T. Watson <sip:xxx@10.1.3.96>
Call-ID: 31416@c.bell-tel.com
Cseq: 1 INVITE

b) Badane jest reakcja SIP UA w gdy w wiadomości podana jest nieistniejąca wersja protokołu SIP.

Spodziewana reakcja: przesłanie odpowiedzi o nieznannej wersji protokołu SIP.

INVITE sip:woytek@10.1.3.96:5062 SIP/7.0
Via: SIP/2.0/UDP 10.1.3.96
From: A. Bell <sip:xxx@10.1.3.96>
To: T. Watson <sip:xxx@10.1.3.96>
Call-ID: 31417@c.bell-tel.com
Cseq: 1 INVITE

TEST 1_8

Badane jest reakcja SIP UA w gdy w metodzie INVITE występuje nieznanany schemat autoryzacji.

Spodziewana reakcja: odrzucenie żądania.

```
INVITE sip:woytek@10.1.3.96:5062 SIP/2.0
To: sip:xxx@10.1.3.96:5060
From: sip:xxx@10.1.3.96:5060
Call-ID: 0ha0isndaksdj@10.0.1.1
Cseq: 8 INVITE
Via: SIP/2.0/UDP 135.180.130.133
Authorization: Super-PGP ajsohdaosdh0asyhdaind08yasdknasd09asidhas0d8
```

Dodatek B

Testy Konfiguracji 2:

TEST 2_1

Wysłanie do każdego z komunikujących się SIP UA metody NEWMETHOD.

Spodziewana reakcja: odrzucenie żądania bez wpływu na trwające połączenie.

```
NEWMETHOD sip:woytek@maniana:5062 SIP/2.0
To: sip:xxx@10.1.3.96:5060
From: sip:xxx@10.1.3.96:5060
Call-ID: 1110ha0isndaksdj@10.1.1.1
CSeq: 1 NEWMETHOD
Via: SIP/2.0/UDP 135.180.130.133
Expires: 7200
```

TEST 2_2

Wysłanie do każdego z komunikujących się SIP UA metody ACK (zawsze istnieje para INVITE, ACK).

Spodziewana reakcja: odrzucenie żądania bez wpływu na trwające połączenie.

```
ACK sip:woytek@10.1.3.96:5062 SIP/2.0
To: sip:xxx@10.1.3.96:5060
From: sip:xxx@10.1.3.96:5060
Call-ID: 1110ha0isndaksdj@10.1.1.1
CSeq: 1 ACK
Via: SIP/2.0/UDP 135.180.130.133
Expires: 7200
```

TEST 2_3

Wysłanie do każdego z komunikujących się SIP UA odpowiedzi 180 Ringing.

Spodziewana reakcja: bez wpływu na trwające połączenie.

```
SIP/2.0 180 Ringing
To: sip:woytek@10.1.3.96:5062
From: sip:xxx@10.1.3.96:5060
Call-ID: 1110ha0isndaksdj@10.1.1.1
CSeq: 1 INVITE
Via: SIP/2.0/UDP 135.180.130.133
```

TEST 2_4

Wysłanie do każdego z komunikujących się SIP UA odpowiedzi OK.

Spodziewana reakcja: bez wpływu na trwające połączenie.

```
SIP/2.0 OK
To: sip:woytek@10.1.3.96:5062
```


From: sip:xxx@10.1.3.96:5060
Call-ID: 1110ha0isndaksdj@10.1.1.1
CSeq: 1 INVITE
Via: SIP/2.0/UDP 135.180.130.133

TEST 2_5

Wysłanie do każdego z komunikujących się SIP UA metody CANCEL.

Spodziewana reakcja: bez wpływu na trwające połączenie.

CANCEL sip:woytek@10.1.3.96:5062 SIP/2.0
To: sip:xxx@10.1.3.96:5060
From: sip:xxx@10.1.3.96:5060
Call-ID: 1110ha0isndaksdj@10.1.1.1
CSeq: 1 CANCEL
Via: SIP/2.0/UDP 135.180.130.133

TEST 2_6

Wysłanie do każdego z komunikujących się SIP UA metody BYE.

Spodziewana reakcja: bez wpływu na trwające połączenie.

BYE sip:woytek@10.1.3.96:5062 SIP/2.0
From: sip:xxx@10.1.3.96:5061
Call-ID: 111daasadasx@10.1.1.1
CSeq: 8 BYE
Via: SIP/2.0/UDP 135.180.130.133
Content-Length: 0

TEST 2_7

Wysłanie do każdego z komunikujących się SIP UA metody INVITE, tak aby sprawdzić czy testowana aplikacja jest w stanie bez przeszkód obsłużyć więcej niż jedno połączenie.

Spodziewana reakcja: bez wpływu na trwające połączenie.

INVITE sip:sip:woytek@10.1.3.96:5062 SIP/2.0
Via: SIP/2.0/UDP 135.180.130.133
CSeq: 0 INVITE
Call-ID: 198asdh@10.1.1.12
From: sip:xxx@10.1.3.96:5060
To: sip:xxx@10.1.3.96:5060
Content-Type: application/sdp

v=0
o=mhandley 29739 7272939 IN IP4 126.5.4.3
c=IN IP4 135.180.130.88
m=audio 49210 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC/8000

Literatura

- [1] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg – „SIP: Session Initiation Protocol” – Request for Comments nr. 3261 lipiec 2002
- [2] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg – „SIP: Session Initiation Protocol” – Request for Comments nr. 2543 marzec 1999
- [3] S. Salsano, L. Veltri, D. Papalilo – ”SIP Security Issues: The SIP Authentication Procedure and its Processing Load” – IEEE Network, vol. 16, vol. 6, November 2002
- [4] W. Stallings - “Cryptography and Network Security : Principles and Practice, Second Edition”. Prentice-Hall, June 1998.
- [5] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart. “HTTP Authentication : Basic and Digest Access Authentication”. Request For Comments 2617. Internet Engineering Task Force, June 1999.
- [6] T. Dierks, C. Allen. “The TLS protocol version 1.0”. Request For Comments 2246. Internet Engineering Task Force, January 1999.
- [7] R. Rivest. “The MD5 Message-Digest Algorithm”. Request For Comments 1321. Internet Engineering Task Force, April 1992.
- [8] T. Berners-Lee, R. Fielding, H. Frystyk. “Hypertext Transfer Protocol -- HTTP/1.0”. Request For Comments 1945. Internet Engineering Task Force, May 1996.
- [9] N. Borenstein, N. Freed. “MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies “. Request For Comments 1341. Internet Engineering Task Force, June 1992.
- [10] B. Ramsdell. “S/MIME Version 3 Message Specification”. Request For Comments 2633. Internet Engineering Task Force, June 1999.
- [11] J. Callas, L. Donnerhacke, H. Finney, R. Thayer, “Open PGP Message Format”. Request For Comments 2440. Internet Engineering Task Force, November 1998.
- [12] P. Gajowniczek, M. Średniawa - „Voice over IP – Wykorzystanie techniki IP do przesyłania głosu” - CITCOM-PW październik 1999
- [13] H. Sinnreich, A. Johnston – „Internet Communications Using SIP“ – Wiley Computer Publishing

Artykuł recenzowany.