Warszawa, kwiecień 2019 r.

Politechnika Warszawska
Wydział Elektroniki i Technik Informacyjnych

Bezpieczeństwo Systemów i Sieci

Instrukcja do zajęć laboratoryjnych nt. GnuPG

opracował dr inż. Jacek Wytrębowicz

Spis treści

Wprowadzenie 2
Informacje uzupełniające 4
Użyteczne polecenia terminalowe gpg2 8
Potencjalnie użyteczne linki 9
Potencjalne pytania sprawdzające 10
Przygotowanie narzędzi do ćwiczenia 11
Podpowiedzi przydatne w czasie realizacji ćwiczenia 14
Oczekiwane rezultaty ćwiczenia 14
Przebieg ćwiczenia 15

Wprowadzenie

Celem zajęć jest praktyczne zaznajomienie się z narzędziami kryptograficznymi GNU Privacy Guard (GnuPG) służącymi do uwierzytelniania oraz do bezpiecznej wymiany plików i poczty elektronicznej.

W ramach przygotowania do ćwiczenia konieczne jest uważne przeczytanie niniejszej instrukcji. Na zajęcia można zabrać własną pamięć USB w celu zapisania na niej stworzonych przez siebie kluczy.

GnuPG jest swojego rodzaju następcą narzędzia Pretty Good Privacy (PGP). PGP był programem napisanym w 1991 r. przez Phila Zimmermanna. Narzędzie to, rozwijane do dziś dzień, było przez szereg lat dostępne bezpłatnie. Jego komercjalizacja spowodowała rozwój otwartych narzędzi alternatywnych, z których popularność zdobył GnuPG. Obydwa narzędzia PGP i GnuPG są zgodne z ze standardem „OpenPGP Message Format" (RFC 4880). Przez wiele lat GnuPG rozpowszechniane było jedynie w wersji z terminalowym interfejsem użytkownika. Od kilku lat zaczęły pojawiać się okienkowe nakładki dla GnuPG na różne systemy operacyjne, oraz wtyczki do popularnych programów poczty elektronicznej.

Powszechnie dostępna wersja terminalowa GnuPG, działająca na wszystkich popularnych systemach operacyjnych, to *gpg2*. Na serwerach i systemach wbudowanych zwykle istnieje nieco prostsza wersja, to jest *gpg*. W niektórych instalacjach *gpg* istnieje jako skrót do *gpg2*. Aplikacje graficzne służące do zarządzania kluczami, będące ograniczonymi interfejsami do funkcji *gpg2*, to:

- *KGpg* dla systemu Linux. Jest ono wbudowane w aplikację KDE, patrz: https://utils.kde.org/projects/kgpg/. Dokumentację narzędzia można znaleźć pod adresem: https://docs.kde.org/stable/en/kdeutils/kgpg/index.html.
- *Gpg4win* dla systemu Windows. Narzędzie to wraz z dokumentacją można pobrać ze stron: http://www.gpg4win.org/about.html. W pakiecie znajduje się wersja terminalowa *gpg2* i okienkowa *Kleopatra*.
- *GPG Keychain.app* dla systemu Mac OS X. Narzędzie to wraz z dokumentacją można pobrać ze stron: https://gpgtools.org.

Na zajęciach laboratoryjnych używać będziemy narzędzie terminalowe gpg2, oraz klienta pocztowego *Thunderbird* z zainstalowaną wtyczką *Enigmail*. O szyfrowaniu poczty w *Thunderbird* można przeczytać np. tu: http://sekurak.pl/szyfrowanie-poczty-w-thunderbird/. Serwer pocztowy zalecany do używania w czasie zajęć to mion.elka.pw.edu.pl. Chętni mogą używać również narzędzie graficzne KGpg – lecz uwaga, jego uproszczony interfejs użytkownika jest nieco ograniczony funkcjonalnie.

Bezpieczne usuwanie plików i szyfrowanie dysków nie są zagadnieniami tytułowymi tych zajęć. Trudno też dla nich zaproponować ciekawe ćwiczenia praktyczne. Jednak, ze względu na ich wagę dla bezpieczeństwa danych, są one tu poruszane.

Student przystępujący do zajęć musi posiadać podstawową wiedzę na następujące tematy:

- algorytmy szyfrowania symetrycznego,
- algorytmy szyfrowania asymetrycznego,
- funkcje skrótu,
- podpis cyfrowy,
- koperta cyfrowa,
- bezpieczne usuwanie plików (namawiam na przejrzenie *man shred*).

Ponadto oczekuję encyklopedycznej wiedzy na temat serwerów kluczy; polecam http://en.wikipedia.org/wiki/Key_server_(cryptographic).

Uwaga: *Certyfikat* w powszechnym znaczeniu to oficjalny dokument potwierdzający zgodność z normami, spełnianie wymogów, autentyczność i inne; zaświadczenie lub świadectwo. W kontekście podpisów elektronicznych jest to złożona struktura (pamiętana w pamięci masowej, przesyłana w sieci, albo przetwarzana) zawierająca rekordy danych i rekordy uwierzytelniające.

W wielu dokumentach informatycznych, jak również w niniejszej instrukcji, termin *certyfikat* jest używany wieloznacznie, tj. jako struktura danych zawierająca klucz publiczny, zbiór kluczy prywatnych i informacje opisujące właściciela klucza, albo jako struktura danych zawierająca jedynie klucz publiczny i informacje opisujące właściciela. Wieloznaczność ta upraszcza opis. Choć warto zapamiętać, że przez certyfikat klucza publicznego rozumiemy strukturę zawierającą: klucz publiczny podmiotu, opis tożsamości podmiotu, podpis cyfrowy złożony przez trzecią stronę na dwóch powyższych informacjach.

Informacje uzupełniające

Niniejszy rozdział zawiera skrótowe przypomnienie wybranych treści prezentowanych na wykładzie, dodatkowe informacje przydatne w niniejszych zajęciach laboratoryjnych, oraz ciekawostki. Pokrótce omówione są:

- podpis cyfrowy;
- koperta cyfrowa;
- szyfrowanie konwencjonalne;
- klucze dzielone;
- podpis grupowy;
- standard OpenPGP;
- protokoły wymiany kluczy GnuPG;
- dobre praktyki dotyczące bezpieczeństwa kluczy.

Podpis cyfrowy. Uwaga terminy: podpis cyfrowy, podpis elektroniczny, poświadczenie elektroniczne, certyfikat i zaświadczenie certyfikujące są definiowane w kontekstach prawnych, których tutaj nie będziemy analizować. Ciekawostką może być to, że są systemy prawne wymagające aby do szyfrowania i podpisywania były używane odrębne pary kluczy. Z technicznego punktu widzenia, przez podpis cyfrowy rozumiemy uzupełnienie dokumentu cyfrowego pozwalające zapewnić autentyczność pochodzenia, niezaprzeczalność utworzenia i integralność danego dokumentu. Podpis ten jest odpowiednikiem podpisu własnoręcznego. Jego podrobienie, przy stosowaniu dobrych kryptosystemów i zalecanych długości klucza, jest praktycznie niemożliwe. Jest zatem znacznie trudniej podrobić podpis cyfrowy aniżeli własnoręczny. W sprawdzaniu ważności podpisu istotna jest data jego wystawienia – późniejsze unieważnienie klucza nie wpływa na ważność tego podpisu. Do tworzenia podpisów cyfrowych konieczne jest używanie powszechnych systemów uwierzytelniania bazujących na kryptografii asymetrycznej. Są to: hierarchia urzędów certyfikujących PKI (standard X.509) oraz zdecentralizowana sieć zaufania (standard OpenPGP).

W uproszczeniu podpis cyfrowy jest skrótem cyfrowym danego dokumentu zaszyfrowanym prywatnym kluczem autora/nadawcy. Weryfikacja podpisu polega na sprawdzeniu zgodności pomiędzy wyliczonym skrótem odebranego dokumentu a skrótem uzyskanym z podpisu zdeszyfrowanego kluczem publicznym nadawcy. Stosowany algorytm podpisu cyfrowego jest nieco bardziej skomplikowany. Stosuje on losowy parametr, generując dla tego samego dokumentu inną wartość podpisu przy każdej kolejnej iteracji podpisywania.

Koperta cyfrowa. Termin ten nie ma jednej standardowej definicji. W różnych publikacjach może on mieć nieco różniące się znaczenia. Najpowszechniej jest on używany w znaczeniu szyfrowania hybrydowego; w którym losowo generowany klucz symetryczny, użyty do zaszyfrowania długiego dokumentu, jest następnie szyfrowany kluczem publicznym odbiorcy. Odbiorca używając swój klucz prywatny deszyfruje klucz symetryczny, którym odszyfrowuje dokument. Dzięki szyfrowaniu hybrydowemu efektywnie przeprowadzamy operacje szyfrowania i deszyfrowania długiego dokumentu, oraz bezpiecznie przekazujemy tymczasowy klucz. Znacznie trudniejszym staje się też łamanie szyfru, gdyż potencjalny agresor nie może zebrać dużej ilości próbek szyfrowanych tym samym kluczem.

Termin koperty cyfrowej może również oznaczać podobny schemat do powyższego, ale z użyciem jedynie kryptografii symetrycznej. W tym przypadku tymczasowy klucz użyty do szyfrowania dokumentu jest szyfrowany wcześniej uzgodnionym kluczem

symetrycznym. Skutecznie utrudniamy w ten sposób złamanie uzgodnionego klucza – agresor może zebrać małą liczbę próbek, ponadto nie zawierają one przewidywalnych treści.

Niektórzy autorzy definiują termin koperty cyfrowej szerzej. Przyjmują oni, że kluczem tymczasowym szyfrowany jest nie tylko dokument ale również jego funkcja skrótu albo podpis dokumentu uwierzytelniający jego nadawcę.

Szyfrowanie konwencjonalne. Termin ten często oznacza po prostu kryptografię symetryczną. Jest on też używany w znaczeniu koperty cyfrowej z prostym kluczem symetrycznym, zwykle frazą hasłową. Znaczenie to rozpowszechnione zostało wraz z darmowym w minionych czasach programem PGP. Istota szyfrowania konwencjonalnego polega na wygenerowaniu silnego klucza tymczasowego, zaszyfrowania nim długiego dokumentu, a następnie zaszyfrowania samego klucza prostą frazą tekstową. Frazę tę łatwo możemy przekazać telefonicznie odbiorcy, który zdeszyfruje nią klucz, a następnie tym kluczem odebrany dokument. Bezpieczeństwo tego mechanizmu jest znacznie wyższe od powszechnie stosowanego szyfrowania programami archiwizującymi, np. WinZip. Szyfrowanie konwencjonalne jest możliwe przy użyciu narzędzia *gpg2*.

Klucze dzielone. Istotą dzielenia klucza (ang. secret sharing, key sharing, key splitting) jest umożliwienie podpisywania dokumentów przez podgrupę osób dzielących dany klucz. Jest to funkcja będąca odzwierciedleniem wymogu prawnego obowiązującego w wielu przedsiębiorstwach, gdzie dla ważności danego dokumentu (zobowiązania) musi on być podpisany przez określoną liczbę osób z grupy upoważnionych, np. przez 2 z 5 członków zarządu firmy. Mechanizm ten chroni też wiadomość przed utratą zabezpieczającego ją klucza. Specyfika zastosowania tego mechanizmu spowodowała, że jest on dziś obecny jedynie w komercyjnych aplikacjach, np. PGP; choć można znaleźć przydatne darmowe biblioteki dla programistów, np. *libgfshare*.

Schemat dzielenia klucza jest następujący: Generujemy zwykła parę kluczy. Dzielimy klucz prywatny na potrzebną ilość części, przyznając je różnym osobom (jedna osoba może otrzymać więcej niż jedną część). Definiujemy minimalną liczbę części potrzebnych do podpisywania dokumentów. Klucze dzielone, podobnie jak zwykły klucz prywatny, używane są zarówno do podpisywania własnych dokumentów jak i do deszyfrowania dokumentów odbieranych i zaszyfrowanych publiczną parą klucza.

Możliwy jest również scenariusz użycia tego mechanizmu przez jednego użytkownika, który dzieli klucz prywatny pomiędzy 3 swoje urządzenia, tak aby dwa były niezbędne do wykonania operacji podpisu lub deszyfrowania (ewentualnie do odtworzenia klucza prywatnego). Rozwiązanie to chroni użytkownika przed skutkami utraty lub awarii jednego z tych urządzeń.

Należy zauważyć, że klucze przechowywane w częściach są trudniejsze do zdobycia, przez potencjalnego atakującego, niż klucze trzymane w całości. Każda część jest przecież przechowywana w innym chronionym miejscu i zabezpieczona odrębnym hasłem.

Podpis grupowy. Istotą podpisu grupowego jest zachowanie anonimowość osób podpisujących należących do danej grupy. Przykładem zastosowania tego mechanizmu jest gwarantowanie, że otrzymany dokument podpisany był przez pracownika danego przedsiębiorstwa bez możliwości odkrycia, który to pracownik. Inny przykład, to karty

dostępu przydzielane pracownikom bez możliwości rejestrowania i śledzenia ruchu poszczególnych osób (ze względów prawnych lub etycznych). W rozwiązaniach tego typu może występować pozycja menadżera grupy, pozycja menadżera odwoływania członków grupy. W rozwiązaniach z menadżerem, może on złamać anonimowość (ujawnić podpisującego).

Standard OpenPGP. Termin OpenPGP oznacza oprogramowanie służące do szyfrowania i podpisywania, z użyciem mechanizmów certyfikatów, zarządzania kluczami, koperty cyfrowej, kompresji i konwersji (Radix-64) wiadomości. Standard OpenPGP (RFC 4880) definiuje struktury danych i struktury wymienianych wiadomości. Poniżej podanych jest kilka wybranych z RFC 4880 informacji, w celu zilustrowania istoty tego dokumentu.

Wiadomości, pęki kluczy (ang. keyrings), certyfikaty i inne struktury zbudowane są z rekordów nazywanych pakietami. Każdy pakiet zawiera nagłówek (zmiennej długości) i treść. Nagłówek określa typ pakietu, długość nagłówka i długość pakietu. Pakiety mogą zawierać pakiety. Oto kilka przykładowych typów pakietów:

- Signature Packet,
- Symmetric-Key Encrypted Session Key Packet,
- One-Pass Signature Packet,
- Secret-Key Packet,
- Public-Key Packet,
- Secret-Subkey Packet,
- Compressed Data Packet,
- Symmetrically Encrypted Data Packet.

RFC 4880 definiuje identyfikatory dla standardowych i eksperymentalnych (nowych) algorytmów szyfrowania asymetrycznego i symetrycznego jak również dla funkcji skrótu i algorytmów kompresji. Konwersja Radix-64 obejmuje kodowanie MIME base64 oraz wyliczenie CRC-24.

Wiadomości szyfrowane są jednorazowym kluczem symetrycznym. Klucz ten po zaszyfrowaniu go publicznym kluczem odbiorcy dodawany jest na początku zaszyfrowanej wiadomości. Możliwe jest stosowanie szyfrowania konwencjonalnego zamiast kryptografii asymetrycznej. Jeżeli wiadomość ma być podpisana, to wyliczony podpis dołączany jest za wiadomością i razem z nią szyfrowany. Oczywiście możliwe jest wysyłanie jedynie podpisanych wiadomości bez szyfrowania.

Kompresja jest zalecana, gdyż jej efektem ubocznym jest znaczne utrudnienie przeprowadzenia ataku. Kompresowana jest wiadomość po wyliczeniu podpisu lecz przed szyfrowaniem.

Protokoły wymiany kluczy GnuPG. Kiedyś powszechne były LDAP i HTTP. Dziś HKP (port 11371) i HKPS (ten sam port co HTTPS, tj. 443). Do łask wraca LDAP. Pojawiają się propozycje rejestrowania kluczy w rejestrach typu blockchain.

Dobre praktyki dotyczące bezpieczeństwa kluczy:

- generować je o maksymalnie możliwej długości, gdyż im krótszy klucz, tym łatwiej go złamać metodą „ataku siłowego”;

- generować je z ograniczonym czasem ważności, ze względu na rosnącą moc obliczeniową maszyn i postęp technologiczny;
- przechowywać klucz prywatny na własnej pamięci (np. USB lub dyskowej), którą możemy łatwo chronić przed niepowołanym dostępem;
- kopie bezpieczeństwa tej pamięci wykonywać na urządzeniach lokalnych i odpowiednio je chronić;
- generować i chronić w bezpiecznym miejscu (miejscach) certyfikat unieważniania, aby móc odwołać skompromitowany lub utracony klucz prywatny, albo gdy zapomnimy jego hasło dostępu.

Z wygenerowaną parą kluczy można związać więcej aniżeli jeden adres e-mail, co pozwala na podpisywanie poczty tym samym kluczem gdy używamy kilku adresów.

Warto tu wspomnieć, że niektórzy użytkownicy OpenGPG drukują odcisk swego klucza na kartach wizytowych. Istnieją też tacy, którzy promują zamiast wydruku tekstowego stosowanie zapisu graficznego *QR Code*. Pozwala to na szybką i bezbłędną wymianę odcisków klucza z użyciem smartfonów i laptopów. Trzeba przyznać zapis na wizytówce np. „8E8A BE95 1B11 F74A F804 F7A6 601E FBBA 760C B092" nie jest dekoracyjny ani łatwo czytelny, w przeciwieństwie do:

Użyteczne polecenia terminalowe gpg2

Uzyskanie szybkiej pomocy	gpg2 --help
Generowanie pary kluczy	gpg2 --gen-key
Wyświetlenie kluczy prywatnych	gpg2 --list-secret-keys
Wyświetlenie odcisku klucza	gpg2 --fingerprint <Your_Key_ID>
Wygenerowanie certyfikatu unieważniania	gpg2 --output <revcert.asc> --gen-revoke <key_id>
Użycie certyfikatu unieważniania[1]	gpg2 --import <revcert.asc>
Edycja parametrów klucza	gpg2 --edit-key <Your_Key_ID>
Usunięcie klucza prywatnego	gpg2 --delete-key <Your_Key_ID>
Wysłanie klucza na serwer	gpg2 --keyserver <keyserver> --send-key <Your_Key_ID>
Pobranie klucza z serwera	gpg2 --keyserver <keyserver> --recv-keys <Key_ID>
Podpisanie czyjegoś klucza	gpg2 --sign-key <Key_ID>
Podpisanie pliku	gpg2 --sign <fname> gpg2 --clearsign <fname> gpg2 --detach-sign <fname> gpg2 --detach-sign --armor <fname>
Zaszyfrowanie pliku	gpg2 --encrypt<fname>
Sprawdzenie podpisu pliku	gpg2 --verify <sigfile> [<orgfile>]
Odszyfrowanie pliku	gpg2 --decrypt<fname>

Polecam otworzyć okienko terminala i wywołać w nim: *gpg2 -help*; w celu przejrzenia najpopularniejszych opcji. Pełną listę i ich opis znajdziemy w *man gpg2*. Uwaga: *gpg2* można wywoływać bez opcji podając jako argument nazwę pliku; wówczas wykonywana jest akcja stosowna do typu pliku, np. deszyfrowanie pliku *xxx.enc*, weryfikacja podpisu pliku *xxx.sig*, wylistowanie kluczy z pliku *xxx.gpg*. Ponadto, aby odwoływać się do odpowiednich kluczy można używać *short ID* klucza w postaci np."760CB092". Użyteczną opcją jest zapis wyniku do pliku o zadanej nazwie a nie na *stdout*, tj.: -o nazwa_pliku. Wyczerpującą dokumentację *gpg2* można znaleźć pod adresem: https://www.gnupg.org/documentation/manuals/gnupg/.

[1]/ Unieważniany certyfikat musi znajdować się w lokalnym pęku kluczy. Jeżeli go straciliśmy, to musimy pobrać go z serwera kluczy a po unieważnieniu wyeksportować z powrotem.

Potencjalnie użyteczne linki

OpenPGP JavaScript Implementation http://openpgpjs.org .

Wyszukiwanie kluczy publicznych http://keys.gnupg.net.

GnuPG Mini Howto
http://www.dewinter.com/gnupg_howto/english/GPGMiniHowto.html .

The Keysigning Party HOWTO
http://www.cryptnet.net/fdp/crypto/keysigning_party/en/keysigning_party.html .

ProtonMail Homepage https://protonmail.ch

ProtonMail co-founder speech
http://ideas.ted.com/why-we-should-all-care-about-encryption-really/ .

The OpenPGP Homepage http://openpgp.org

OpenPGP dla różnych systemów operacyjnych:

Linux	https://ssd.eff.org/en/module/how-use-pgp-linux
Windows	https://ssd.eff.org/en/module/how-use-pgp-windows-pc
Mac OS X	https://ssd.eff.org/en/module/how-use-pgp-mac-os-x
Mac OS X	https://itunes.apple.com/app/ipgmail/id430780873?mt=8
Mac OS X	https://gpgtools.tenderapp.com/kb/how-to/first-steps-where-do-i-start-where-do-i-begin-setup-gpgtools-create-a-new-key-your-first-encrypted-mail
Android	https://play.google.com/store/apps/details?id=org.sufficientlysecure.keychain https://www.openkeychain.org/about/

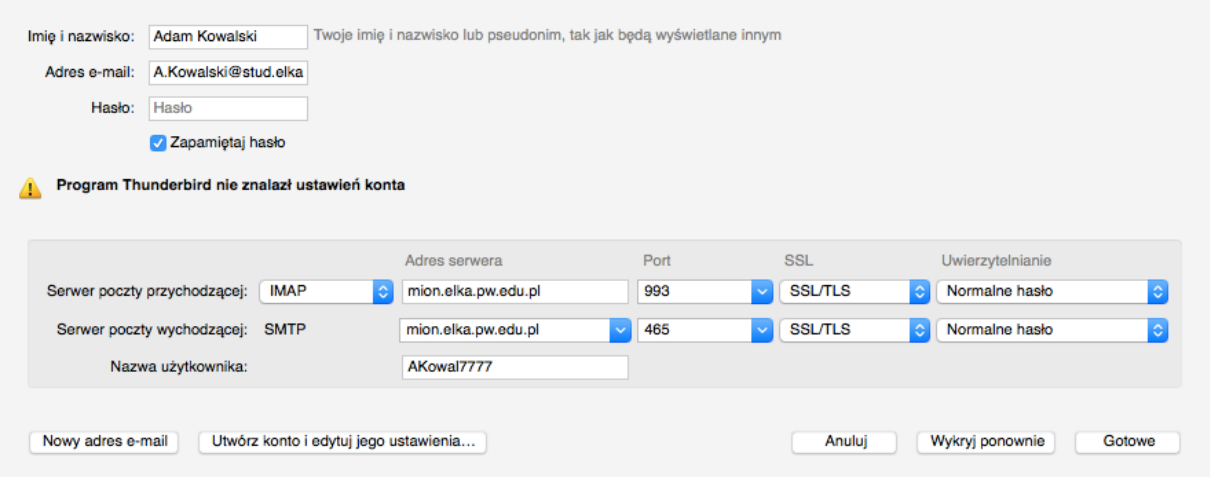
Potencjalne pytania sprawdzające

1. Co zapewnia i co zawiera podpis cyfrowy?
2. Opisać ideę koperty cyfrowej.
3. Opisać ideę szyfrowania konwencjonalnego w pakiecie PGP.
4. Opisać ideę klucza dzielonego.
5. Na czym polega istota klucza grupowego?
6. Dlaczego tworzymy certyfikat unieważniania (revocation certificate | revocation signature) i dlaczego powinniśmy przechowywać go w innym bezpiecznym miejscu aniżeli komputer z naszym podstawowym pękiem kluczy?
7. Jakie mogą być negatywne skutki wykradzenia klucza unieważnienia?
8. Czy mając klucz prywatny możemy unieważnić związany z nim certyfikat z kluczem publicznym bez certyfikatu unieważnienia?
9. Dlaczego na serwerach kluczy przechowywane są unieważnione klucze?
10. W jakim celu podpisujemy klucze publiczne innych użytkowników? Czy funkcja podpisująca używa nasz klucz prywatny czy publiczny?
11. Na czym polega tzw. bezpieczne kasowanie plików?
12. Czy bezpieczne kasowanie pliku daje nam 100% pewności, że usunięte dane nie zostaną odzyskane? Dlaczego?
13. Na jakich dyskach nie ma sensu używać poleceń bezpiecznego kasowania plików i dlaczego?
14. Dlaczego certyfikaty publikowane na serwerach kluczy są trudno usuwalne? Innymi słowy, dlaczego przydatne są klucze publiczne mimo wygaśnięcia ich czasu ważności lub ich unieważnienia?
15. Dlaczego przydatne są klucze prywatne mimo wygaśnięcia ich czasu ważności?
16. Czy konieczne jest wysyłanie certyfikatów do więcej niż jednego serwera kluczy? Dlaczego?
17. Czym istotnym różnią się certyfikaty X.509 i OpenPGP?
18. Jak przekazywać pocztą pliki aby nie tylko nie ujawniać ich treści ale również ich nazw?

Przygotowanie narzędzi do ćwiczenia

Należy pamiętać, że jeżeli na danym komputerze generujemy lub przechowujemy klucze prywatne, to powinien być on odpowiednio chroniony przed niepowołanym dostępem. Nie należy przechowywać kluczy prywatnych na dyskach sieciowych, ani na dyskach lokalnych komputerów używanych przez niezaufane osoby (np. komputery w laboratoriach studenckich). Jeżeli student pozostawi swoje klucze na takim dysku, to naraża się na niebezpieczeństwo ich kompromitacji, natomiast jeżeli zachowa je na prywatnej pamięci USB, to sam odpowiada za bezpieczeństwo tego nośnika. Domniemanym katalogiem w którym przechowywane są pęki kluczy jest ***.gnupg***, w katalogu domowym użytkownika. GnuPG można skonfigurować tak aby pęki kluczy przechowywane były na dołączonej pamięci USB.

Należy skonfigurować klienta pocztowego *Thunderbird* do pracy ze swoim kontem pocztowym np. xxx@stud.elka.pw.edu.pl na serwerze mion.elka.pw.edu.pl (przed nazwą serwera nie może być kropki!) używając protokołów IMAP (port 993) i SMTP (port 465 lub 587), uwierzytelnianie SSL/TLS. Uwaga: login na serwerze nie jest tożsamy z nazwą użytkownika z adresu email. Poniżej podane jest przykładowe okienko konfiguracji konta pocztowego. Zalecane jest ograniczenie synchronizacji folderów z korespondencją pomiędzy serwerem pocztowym a dyskiem użytkownika w sieci laboratoryjnej, gdyż może dojść do przekroczenia przydzielonej pamięci! Aby nie dopuścić do tego, należy w konfiguracji Thunderbird/Konfiguracja kont/Synchronizacja ustawić np. "Synchronizuj tylko wiadomości z ostatnich X dni" i "Nie pobieraj wiadomości większych niż Y KB". Warto sprawdzić działanie serwera pocztowego wysyłając list do siebie i wymieniając się listami z sąsiadami.



Następnie należy skonfigurować *Enigmail*. Zaczynamy od wygenerowania pary kluczy i utworzenia własnego certyfikatu. **Nazwij go swoim imieniem i nazwiskiem! Zdefiniuj krótki czas jego ważności, np. 1 miesiąc**! Czas ten będziesz mógł później zmienić. Podaj używany przez ciebie adres e-mail (ten wprowadzony przy konfiguracji *Thunderbird*). Generacja pary kluczy może trwać od ½ do 3 min. W celu przyspieszenia generacji otwórz dowolne okienko edycyjne i energicznie wprowadzaj weń dowolne znaki). Gdy pojawi się okienko „Utworzono nową parę kluczy" zaznacz opcję „Zapisz certyfikat unieważniania".

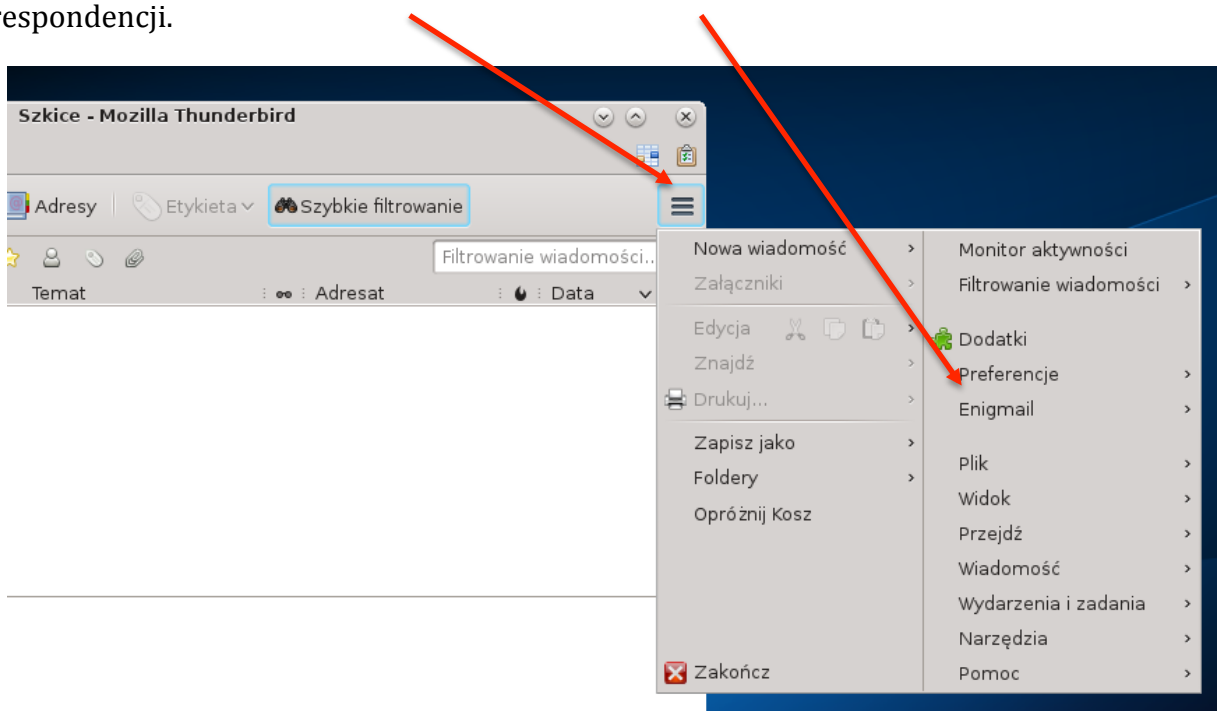
Konfigurując *Enigmail* należy ustawić go do pracy z laboratoryjnym serwerem kluczy (*hkp://bigubu*), predefiniowane serwery najlepiej jest usunąć. Jedynie

laboratoryjny ma być domniemanym serwerem kluczy. Wpisy studentów z serwera *bigubu* będą usunięte tydzień po zajęciach!

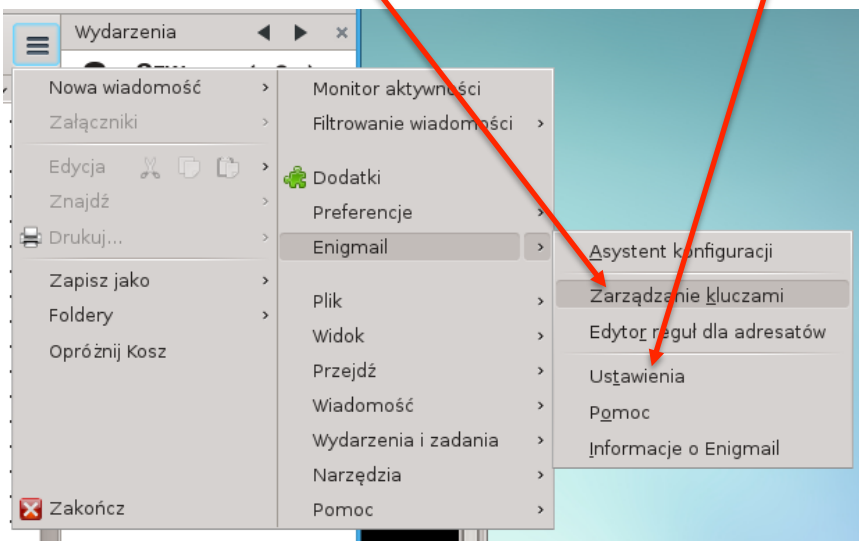
Laboratoryjny serwer kluczy (pełny adres to: *bigubu.ii.pw.edu.pl:11371*) nie synchronizuje się z publicznymi serwerami kluczy i jest niewidoczny z Internetu. Dzięki temu studenci, eksperymentując z nim, nie zaśmiecają serwerów publicznych.

W okienku terminala należy używać parametru *gpg2 --keyserver bigubu ...* . Polecam też otworzyć w przeglądarce WWW stronę http://bigubu:11371.

Aby skonfigurować *Enigmail* uruchamiamy jego *Asystenta konfiguracji*. Jeżeli jeszcze nie utworzyłeś własnego certyfikatu musisz to teraz zrobić. Istotnym parametrem, dla pracy *Enigmail*, jest wybranie swojego klucza jako domniemanego do obsługi korespondencji.



Wtyczka *Enigmail* do programu pocztowego *Thunderbird* oferuje interfejs do zarządzania kluczami. Chcąc z niego korzystać należy wcześniej zdefiniować adres serwera kluczy (*hkp://bigubu*) wywołując okienko *Ustawienia* i wybrać „Wyświetl ustawienia zaawansowane”.



Chcąc korzystać z menadżera kluczy KGpg należy go skonfigurować. Jeżeli w twoim pęku kluczy jeszcze nie ma twojego certyfikatu (mogłeś go wcześniej utworzyć innym narzędziem, np. *gpg2* lub *Enigmail*), to KGpg przy pierwszym uruchomieniu zaproponuje jego utworzenie. Konfigurując KGpg również należy ustawić go do pracy z laboratoryjnym serwerem kluczy (*hkp://bigubu*)

Uwaga: KGpg po pierwszym uruchomieniu pracuje jako demon i nie daje się drugi raz uruchomić. Ponowne otwarcie okienka menadżera certyfikatów wymaga kliknięcia na ikonę KGpg znajdującą się w dolnym rozwijalnym menu. Aby rozwinąć to menu należy kliknąć na ikonę KDE, która ukaże się po kliknięciu z prawej strony w dolnym pasku (jak na rysunku poniżej).

Można jednocześnie korzystać z *gpg2*, menadżera KGpg i menadżera *Enigmail*. Wszystkie te narzędzia operują na tych samych pękach kluczy, tj. plikach w katalogu ~/.gnupg/. Proszę o tym pamiętać, gdyż daje się rozpocząć konfliktowe operacje przy użyciu tych narzędzi.

Uwaga: Możliwe jest komunikowanie się z serwerem *bigubu* z sieci zewnętrznej pod warunkiem posiadania konta na maszynie *galera*. W tym celu należy zestawić tunel, np.:

ssh -L 11371:bigubu:11371 kowalski@galera.ii.pw.edu.pl. Po jego zestawieniu w lokalnej przeglądarce wybieramy http://localhost:11371, zaś w menadżerze kluczy ustawiamy adres serwera kluczy na hkp://localhost.

Podpowiedzi przydatne w czasie realizacji ćwiczenia

- Efekty poleceń wydawanych w czasie terminalowej sesji gpg2 zapisywane są dopiero po jej zamknięciu!
- Większość klientów pocztowych i menadżerów plików ma wbudowane wsparcie dla szyfrowania i podpisywania certyfikatami X.509 (często opisywanymi jako S/MIME). Standardy X.509 i OpenPGP są różne! Zatem do ich używania pojawiają się różne ikonki i pozycje w menu.
- *KGpg* raz uruchomiony, po rozpoczęciu każdej następnej sesji uruchamia się automatycznie. Nie pozwala się uruchomić drugi raz! Ikonka do otwarcia okienka *KGpg* znajduje się w rozwijanym menu dolnego paska narzędziowego.
- Efekty zmian w pęku kluczy wprowadzonych w sesji terminalowej widoczne są w okienku *KGpg* dopiero po jego odświeżeniu (np. klawiszem F5).
- Dopisanie kolejnego adresu e-mail w *KGpg* wymaga kliknięcia prawym klawiszem myszy na swój klucz, pojawi się wówczas menu z którego trzeba wybrać „Dodaj identyfikator użytkownika". Niestety po ostatniej aktualizacji systemu operacyjnego KGpg przestał obsługiwać tę funkcję. Należy więc użyć polecenia terminalowego albo menadżera *Enigmail*.
- *KGpg* nie wyświetla unieważnionych certyfikatów. Aby je zobaczyć trzeba użyć *gpg2* lub menadżera kluczy *Enigmail*.

Oczekiwane rezultaty ćwiczenia

- Na serwerze *bigubu*:
  - certyfikat z ważnym kluczem publicznym,
  - złożony podpis na certyfikacie innego studenta,
  - unieważniony certyfikat z dwoma różnymi adresami email.
- Podpisany i zaszyfrowany list do prowadzącego o wskazanej zawartości (polecenie 12).

Sprawdzając obecność certyfikatów na serwerze wyszukuję je po nazwisku studenta, dlatego nazwa certyfikatu lub adres e-mail powinien zawierać nazwisko.

Przebieg ćwiczenia

Konfigurowanie

1. Skonfiguruj klienta pocztowego i wtyczkę *Enigmail*.
2. Wygeneruj swój certyfikat, jeżeli jeszcze tego nie zrobiłeś. Sprawdź listę serwerów kluczy. Powinien się tam znajdować adres serwera *hkp://bigubu*, wybrany jako domyślny.
3. Sprawdź czy poprawnie skonfigurowałeś *Enigmaila*. Wyślij do siebie podpisany cyfrowo list i obejrzyj jego postać źródłową.

Szyfrowanie i podpisywanie poczty elektronicznej

4. Wymień się z wybranym sąsiadem listami zawierającymi wasze klucze publiczne. Umieśćcie je w swoich pękach kluczy i sprawdźcie zgodność ich odcisków (ang. fingerprint).
5. Wymieńcie się listami zaszyfrowanymi i podpisanymi cyfrowo. Treścią listów mogą być kurtuazyjne pozdrowienia. Uwaga: adres odbiorcy musi się zgadzać z adresem związanym z posiadanym przez nas jego kluczem publicznym.
6. Wymieńcie się zaszyfrowanymi i podpisanymi cyfrowo listami bez załączników. Podejrzyj źródło wiadomości. Określ na jego podstawie: Czy list był najpierw szyfrowany a potem podpisany, czy też na odwrót?
7. Wymieńcie się zaszyfrowanymi i podpisanymi cyfrowo listami zawierającymi dowolny plik tekstowy jako załącznik. Podejrzyj źródło wiadomości. Określ na jego podstawie: Czy widoczna jest nazwa załączonego pliku bez odszyfrowywania listu?

Szyfrowanie i odszyfrowywanie, podpisywanie i sprawdzanie podpisów

Pliki można szyfrować i podpisywać poleceniami gpg2 lub klikając na nie prawym przyciskiem myszy w oknie menadżera plików (dostępność tej drugiej funkcji zależy od posiadanej dystrybucji systemu Linux).

8. Używając gpg2 podpisz (w trybie znakowym!) plik tekstowy zawierający linię dowolnego tekstu, twoje imię, nazwisko i adres email, tak aby
   - podpis był dołączony do pliku,
   - podpis był w odrębnym pliku.

   Porównaj utworzone pliki.
9. Używając gpg2 zaszyfruj plik tekstowy swoim kluczem publicznym a następnie odszyfruj go.
10. Używając gpg2 zaszyfruj (kluczem sąsiada) swój podpisany plik tekstowy. Uwaga: aby zaszyfrować plik/list dla kogoś, musimy mieć jego klucz publiczny w naszym pęku kluczy.
11. Wymieńcie się z sąsiadem utworzonymi plikami, zdeszyfrujcie je i sprawdźcie ich podpisy.
12. Pobierz klucz publiczny prowadzącego (np. z serwera kluczy). Adres prowadzącego: J.Wytrebowicz@ii.pw.edu.pl.

13. Wyślij do prowadzącego zaszyfrowany i podpisany list zawierający: odpowiedzi na pytania z poleceń 6 i 7, oraz odszyfrowany plik otrzymany od kolegi (z polecenia 11). Tytuł listu powinna rozpoczynać fraza „[BSS]".

Wymiana kluczy poprzez pliki

14. Wyeksportuj do plików swoje klucze prywatny i publiczny. Obejrzyj zawartości otrzymanych plików.
15. Obejrzyj strukturę pakietów w plikach z kluczami (polecenie *gpg2 --list-packets <nazwa pliku>*).

Generowanie i usuwanie kluczy oraz wymiana ich poprzez serwer kluczy

16. Wyślij swój certyfikat na serwer kluczy.
17. Wyszukaj na serwerze kluczy certyfikat wybranego sąsiada i pobierz go. Zweryfikuj autentyczność sprawdzając jego skrót z tym co widzi sąsiad na swoim ekranie.
18. Zmodyfikuj stopień „zaufanie do właściciela" pobranego certyfikatu. Można to zrobić menadżerem kluczy *Enigmail*. Z poziomu terminala można to zrobić poleceniem *--edit-key trust*.
19. Podpisz klucz publiczny sąsiada. Następnie wyeksportuj go na serwer kluczy.
20. Wygeneruj drugi certyfikat z innym identyfikatorem niż poprzednio (np. przy pomocy *gpg2 -- gen-key*) i porównaj właściwości obu swoich certyfikatów.
21. Wygeneruj dlań certyfikat unieważniania (np. używając z poziomu terminala polecenia *gpg2 --gen-revoke*.
22. Dodaj do niego inny używany przez ciebie adres e-mail.
23. Wyślij go na serwer kluczy. Obejrzyj swoje certyfikaty na serwerze przeglądarką WWW http://bigubu:11371.
24. Usuń swój drugi certyfikat w lokalnym pęku kluczy.
25. Pobierz z serwera wyeksportowany klucz.
26. Unieważnij go certyfikatem unieważniania i wyślij na serwer kluczy. Obejrzyj swoje klucze na serwerze przeglądarką WWW http://bigubu:11371.

Uwaga: Ćwiczenie można realizować na swoim laptopie przyniesionym na zajęcia, pod warunkiem, że jest na nim zainstalowany *Thunderbird* z wtyczką *Enigmail*.