Zagadnienia bezpieczeństwa w środowisku sieciowym

Podstawowe zagadnienia i wymagania bezpieczeństwa:

poufność

Zabezpieczenie treści przechowywanych i transmitowanych w systemach komputerowych przed dostępem osób niepowołanych.

integralność

Zapewnienie nienaruszalności przechowywanych danych: brak uszkodzeń i kompletność.

dostępność

Zapewnienie funkcjonowania systemów, oraz utrzymania sprawnej komunikacji, pomimo zaburzeń i zakłóceń spowodowanych świadomie, lub nieświadomie (np. w wyniku czyjegoś błędu) przez czynniki trzecie, jak również przez awarie sprzętu i oprogramowania.

uwierzytelnianie

Możliwość stwierdzenia czy użytkownik systemu jest tym za kogo się podaje.

Zagrożenia i linie obrony

Zagrożenia:

- zagrożenia bezpieczeństwa fizycznego: pożar, kradzież, żywioły, itp.
- włamania na serwery: kradzież, malwersacja, sabotaż, przejęcie sterowania obiektami przemysłowymi lub wojskowymi, ataki terrorystyczne, cyberwojna
- włamania na konta użytkowników: przejęcie, zniszczenie, lub ujawnienie danych
- przejęcie kontroli nad grupą komputerów w celu użycia ich do zorganizowanych ataków, stworzenia zagrożenia, szantażu, itp.
  $\Rightarrow$ ataki typu DOS (*denial of service*) - zablokowanie pracy/świadczenia usług

Linie obrony:

- monitorowanie aktywności sieciowej, wykrywanie dziur w zabezpieczeniach, łatanie tych dziur
- monitorowanie aktywności użytkowników, wspieranie ich bezpiecznej pracy
- prewencyjne blokowanie niektórych usług sieciowych (niepotrzebne serwisy, podejrzane adresy)
- szyfrowanie połączeń
- *firewalling* — zapory sieciowe
- duplikacja usług, archiwizacja plików !!!

Terminologia włamywaczy komputerowych

Terminologia stosowana w języku potocznym często rozmija się z terminologią fachową. Na przykład, **haker** (*hacker*) w informatyce oznacza zdolnego programistę, członka zespołu programistycznego. Prasa popularna zawłaszczyła to określenie i nadała mu znaczenie włamywacza komputerowego. W związku z tym wprowadzono nowe słowo **kraker** (*cracker*) na określenie fachowego włamywacza komputerowego.

Krakerów odróżnia się od **skrypciarzy** (*script-kiddies*), którzy są włamywaczami-amatorami, często znudzonymi młodzieżowcami, posiadającymi komputer i wolny czas, oraz dostęp do internetu i narzędzi o dużych możliwościach.

Fachowcy od bezpieczeństwa komputerowego z konieczności zawodowo zajmują się wykrywaniem dziur w zabezpieczeniach, włamań, oraz innych nadużyć, jak również tworzeniem narzędzi zarówno do wykrywania jak i łatania dziur w zabezpieczeniach — całkiem podobnie do zwykłych włamywaczy. Z tego powodu określa się ich w tej roli jako **białe kapelusze** (*white hats*), w odróżnieniu od **czarnych kapeluszy** (*black hats*), którzy robią często podobne rzeczy, ale w złych zamiarach.

Istnieje cały asortyment narzędzi programowych do monitorowania i testowania zabezpieczeń systemów komputerowych, które posiadają **podwójne wykorzystanie**, tzn. mogą być skutecznym narzędziem w rękach obu rodzajów fachowców od bezpieczeństwa komputerowego.

Dygresja — bezpieczne systemy

Pojawia się pytanie, czy nie jest możliwe całkowicie szczelne i skuteczne zabezpieczenie systemów komputerowych, tak aby ataki na nie w ogóle nie były możliwe?

Wbrew pozorom nie jest to całkiem bezsensowne pytanie, i bezpieczne systemy istnieją.

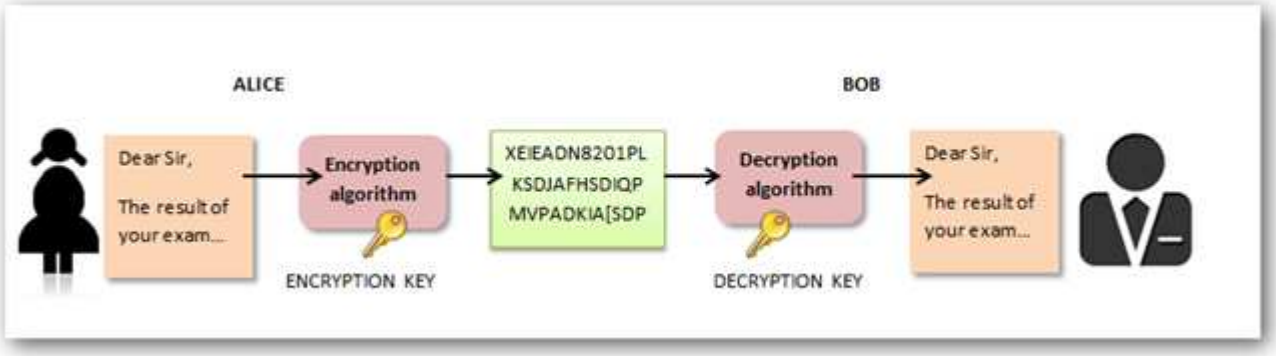
Jak to jest możliwe? Przypomnijmy sobie techniki budowania systemów niezawodnych i wbudowanych. Ważnymi zasadami była oszczędność i minimalizm. Jeśli system komputerowy jest prosty, to kompetentni fachowcy potrafią zbudować go tak by był bezpieczny. Jeśli system przekracza pewien poziom złożoności, to nie da się zidentyfikować i wyeliminować wszystkich jego słabych elementów, i aby uczynić go bezpiecznym, trzeba dodatkowo odizolować go od zagrożeń.

Niestety, dzisiejszy świat nie toleruje minimalizmu, ani ograniczeń w dostępie. Jeśli nawet jakiś system został zaprojektowany jako prosty i niewielki, to jeśli jest przydatny, z czasem będzie rozbudowywany o nowe funkcje. Niekiedy nowe funkcje mają sens, bo powstają lepsze albo innowacyjne rozwiązania. Jednak w większości są nikomu niepotrzebnymi bajerami, które narzuca przemysł i marketing. Ale w projektowanych w szalonym tempie i coraz bardziej złożonych systemach nie da się zachować prostej i logicznej struktury, niezbędnej do zapewnienia niezawodności i bezpieczeństwa.

Podobnie, wymaganie by każdy system był dostępny na wszelkie możliwe sposoby, z wielu urządzeń i przy użyciu różnych technologii, wyklucza bezpieczeństwo.

Szyfrowanie

Jedną z najskuteczniejszych technologii zapewnienia bezpieczeństwa współczesnych systemów komputerowych i sieciowych jest szyfrowanie. Jest stosowane w odniesieniu do pojedynczych dokumentów, jak i ciągłych transmisji sieciowych. Stosowane jest również do uwierzytelniania, zarówno integralności dokumentów jak i tożsamości osób.

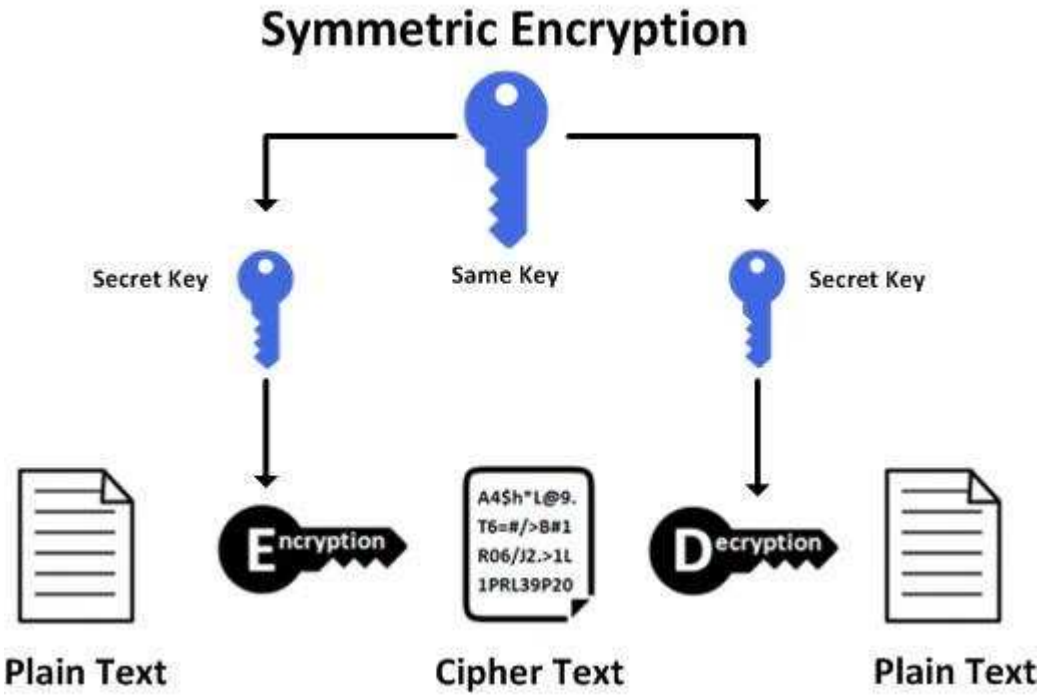


Szyfrowanie było stosowane od starożytności i techniki szyfrowania mają długą historię rozwoju. Obecnie podstawową zasadą w szyfrowaniu jest, że nikt nie dąży do utajnienia algorytmu szyfrowania — te są ogólnie znane i dostępne. Tajny jest jedynie **klucz** szyfrowania, i im bardziej złożony (długi) jest ten klucz, tym trudniej złamać szyfr.

Można to porównać do technologii budowy zamków. Konstrukcje zamków są ogólnie znane, nie tworzy się zabezpieczeń przez wymyślne konstrukcje zamków. Zamiast tego, wybieramy jedną z dobrze znanych, solidnych konstrukcji zamków, oraz unikalny klucz.

Szyfrowanie symetryczne

Najczęściej stosowane są algorytmy szyfrowania **symetrycznego**. Ich istotą jest identyczność kluczy (szyfrów) służących do szyfrowania i deszyfrowania.

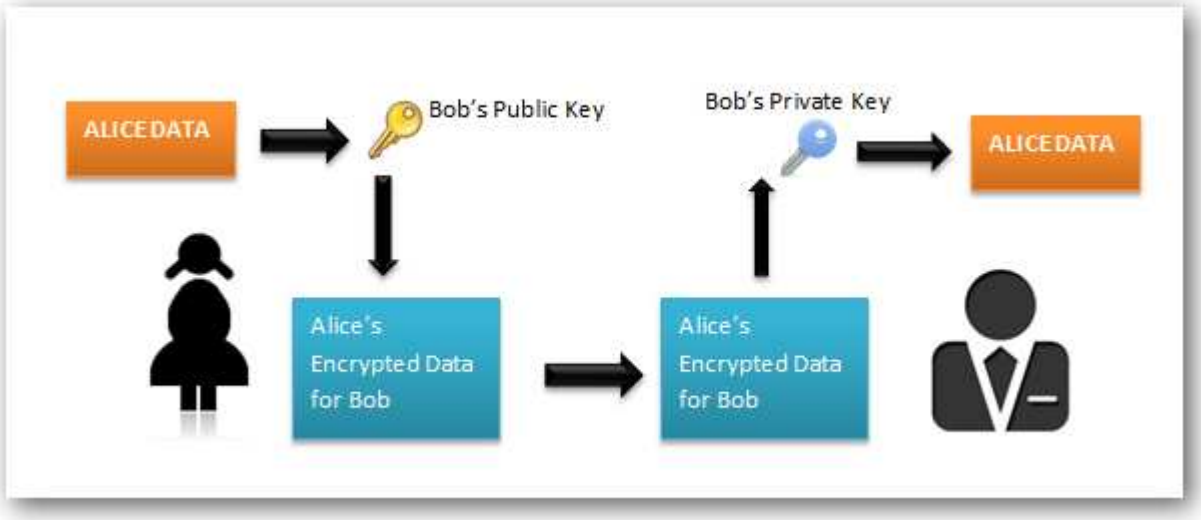


Bezpieczeństwo tego systemu opiera się na tajności kluczy, co stanowi zarazem jego podstawową słabość. W momencie gdy zachodzi konieczność wprowadzenia nowych kluczy, potrzebna jest metoda ich bezpiecznego przekazania sobie przez partnerów.

Jak wiemy ze starych filmów wojennych, spotkanie agentów w celu przekazania nowych szyfrów było jednym z najbardziej niebezpiecznych zadań.

Szyfrowanie asymetryczne

Z powyższego powodu kluczową technologią szyfrowania stosowaną w systemach i sieciach komputerowych jest **szyfrowanie asymetryczne**, zwane również **systemem klucza publicznego**. Klucz szyfrowania każdej jednostki (osoby lub instytucji) składa się z dwóch części: **klucza publicznego**, który jest jawny i może być przesyłany otwartymi kanałami, oraz **klucza prywatnego**, który jest tajny i nigdzie nie wysyłany. Każdy może zaszyfrować wiadomość kluczem publicznym odbiorcy, ale odszyfrować ją będzie mógł tylko właściciel klucza znający jego część prywatną.



Rozwiązuje to podstawową słabość szyfrowania symetrycznego, czyli problem dystrybucji kluczy. Gdy właściciel klucza chce go zmienić, może po prostu wygenerować nową parę kluczy, i dowolnymi kanałami rozpowszechnić nowy klucz publiczny.

Szyfrowanie asymetryczne i symetryczne

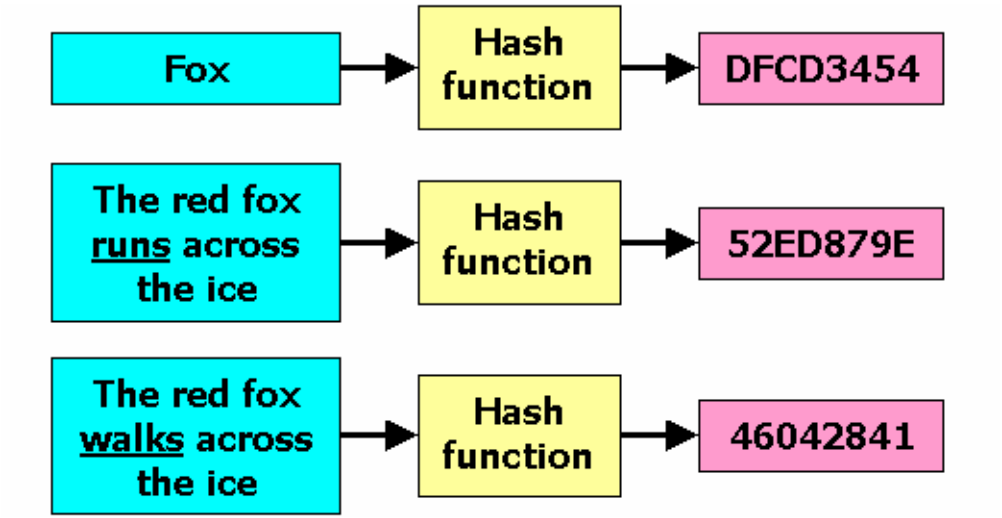
Pytanie: czy pojawienie się metod szyfrowania asymetrycznego powinno spowodować, by starsze metody szyfrowania symetrycznego odeszły do lamusa?

Otóż niekoniecznie. Szyfrowanie asymetryczne jest znacznie mniej efektywne (o kilka rzędów wielkości) niż symetryczne. W praktyce, zwłaszcza duże dokumenty, lepiej jest szyfrować tradycyjnymi szyframi symetrycznymi. Można te dwie metody szyfrowania połączyć w taki sposób, że dokument wysyłany danemu odbiorcy jest szyfrowany szyfrem symetrycznym z jednorazowo wygenerowanym kluczem. Ten klucz zostaje zaszyfrowany kluczem publicznym odbiorcy i wysłany mu razem z dokumentem.

Zauważmy, że rozwiązuje to zarazem problem wysyłania jednego dokumentu wielu odbiorcom. Chcąc zaszyfrować go kluczem publicznym, musiałyby zostać utworzone wielokrotne wersje tego samego dokumentu, każda zaszyfrowana kluczem publicznym kolejnego odbiorcy. Zamiast tego, jest jeden dokument zaszyfrowany szyfrem symetrycznym, i do niego dołączonych wiele kopii klucza symetrycznego (typowo znacznie mniejszego niż sam dokument) zaszyfrowanego kluczami publicznymi poszczególnych odbiorców.

Funkcje mieszające

W systemach informatycznych często przydają się funkcje przypisujące danym argumentom wartości z pewnego zbioru, ale różne dla różnych argumentów. Budowa takich funkcji na ogół polega na silnym mieszaniu różnych fragmentów argumentu (np. bitów), w związku z czym nazywa się je funkcjami **mieszającymi** (*hash functions*).



Własności informatycznych funkcji mieszających: (i) stały, niezależny od argumentu, rozmiar wyniku, (ii) szybkie, deterministyczne obliczanie, (iii) mała liczba kolizji, tzn. jednakowej wartości wyniku dla różnych argumentów. W wielu zastosowaniach informatycznych pewna liczba kolizji jest dopuszczalna, jeśli jest mała w porównaniu z wielkością dziedziny funkcji.

Kryptograficzne funkcje skrótu

Funkcje mieszające znajdują zastosowanie w wielu elementach systemów zabezpieczeń, jednak z tych zastosowań wynikają dodatkowe wymagania dla nich:

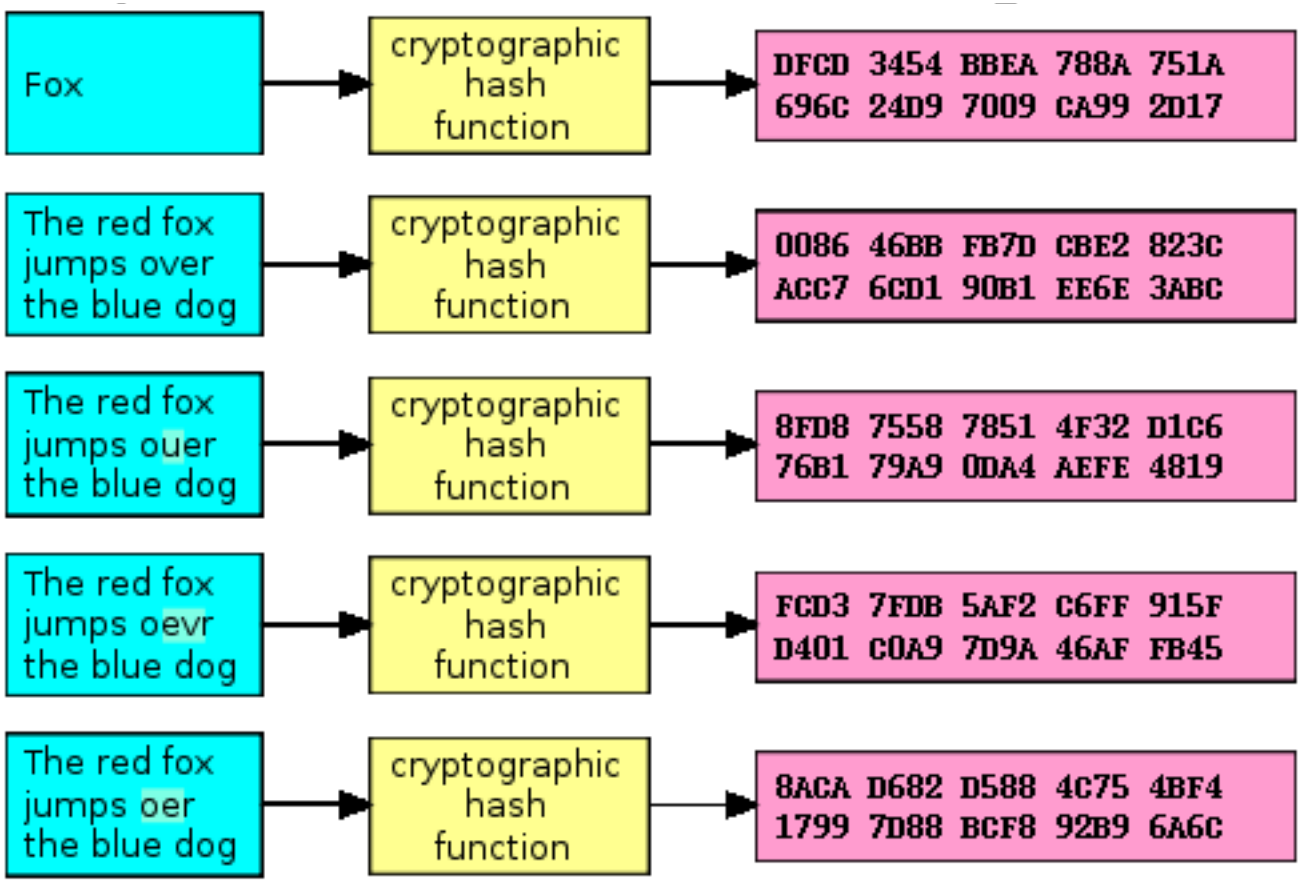
- niemożność odtworzenia jakiejkolwiek części argumentu na podstawie wartości funkcji,
- niezwykle małe prawdopodobieństwo (brak praktycznej możliwości) znalezienia innego argumentu, który dałby tę samą wartość funkcji (tzn. kolizji),
- nawet niewielka zmiana argumentu (np. jednego bitu lub znaku) powinna zmienić wartość funkcji do tego stopnia, by niemożliwe/trudne było wykrycie jakiejkolwiek korelacji między tymi wartościami.

Funkcje mieszające o tych własnościach, stosowane w kryptografii, nazywane są **kryptograficznymi funkcjami skrótu** (*cryptographic hash functions*). Ze względu na wymaganie nieodwracalności stosuje się również określenie funkcji **jednokierunkowych** (*one-way functions*).

Przykładem zastosowania takich funkcji jest przechowywanie haseł. Chcąc zabezpieczyć przechowywane hasła przed możliwością wykradzenia, zamiast przechowywać wersję jawną, możemy przechowywać tylko obliczone skróty, i każdorazowo porównywać je ze skrótem obliczonym z hasła podanego przez użytkownika.

Kryptograficzne funkcje skrótu — przykład

Przykład wartości obliczanych przez popularną funkcję SHA-1:



Ze względu na swoje własności, skróty kryptograficzne można traktować jako swojego rodzaju **sygnatury**, niedwuznacznie identyfikujące oryginalny dokument (i jego konkretną, niezniekształconą wersję).

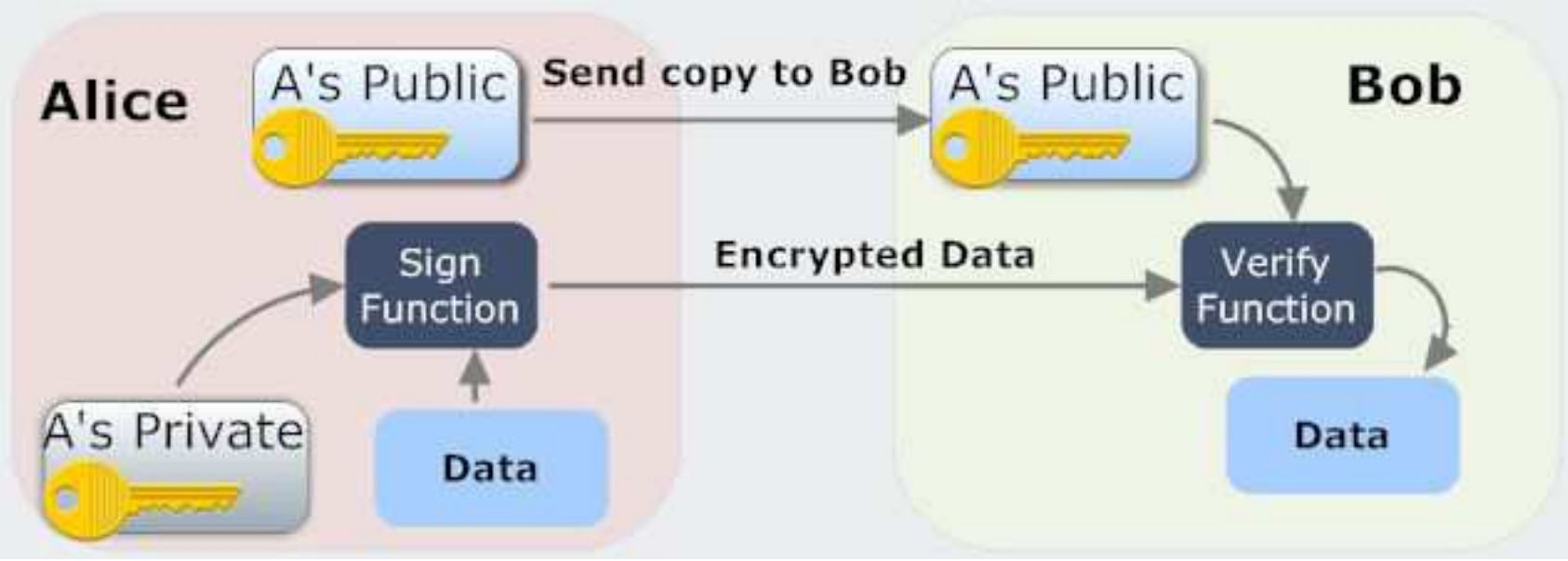
Popularne funkcje skrótów kryptograficznych

Popularny przez wiele lat algorytm skrótu kryptograficznego MD5 generuje ciągi 128-bitowe, często kodowane w postaci 32-znakowych napisów szesnastkowych (heksadecymalnych). Inny popularny algorytm SHA-1 generuje ciągi 160-bitowe, kodowane jako napisy heksadecymalne 40-znakowe.

Jednak kryptografia silnie się rozwija. Istniejące algorytmy są intensywnie badane i poddawane próbom nadużycia, a jednocześnie tworzone są nowe, bezpieczniejsze. Na przykład, w 2011 opublikowano metodę ataku na algorytm SHA-1 pozwalający wygenerować kolizję (alternatywny ciąg bajtów dający tę samą wartość skrótu SHA-1). Metoda wymaga 2^{65} operacji i nikomu nie udało się jeszcze wygenerować takiej kolizji. Pomimo to główni producenci oprogramowania (Microsoft, Google, Mozilla) ogłosili, że od roku 2017 ich systemy nie będą akceptowały certyfikatów opartych na skrótach SHA-1. Istnieje jednak rodzina znacznie bezpieczniejszych algorytmów skrótu SHA-2.

Podpisy cyfrowe

Szyfrowanie kluczem publicznym umożliwia wprowadzenie dodatkowej ważnej funkcji, jaką są **podpisy cyfrowe**. Idea tych podpisów polega na zaszyfrowaniu dokumentu przez nadawcę swoim kluczem prywatnym, i wysłaniu wyniku razem z dokumentem. Zaszyfrowana wersja może być odszyfrowana przez każdego, ale tylko kluczem publicznym nadawcy. Zgodność odszyfrowanego komunikatu z jego pełną wersją dowodzi, że nadawcą jest właściwa osoba, oraz że treść wiadomości jest autentyczna.

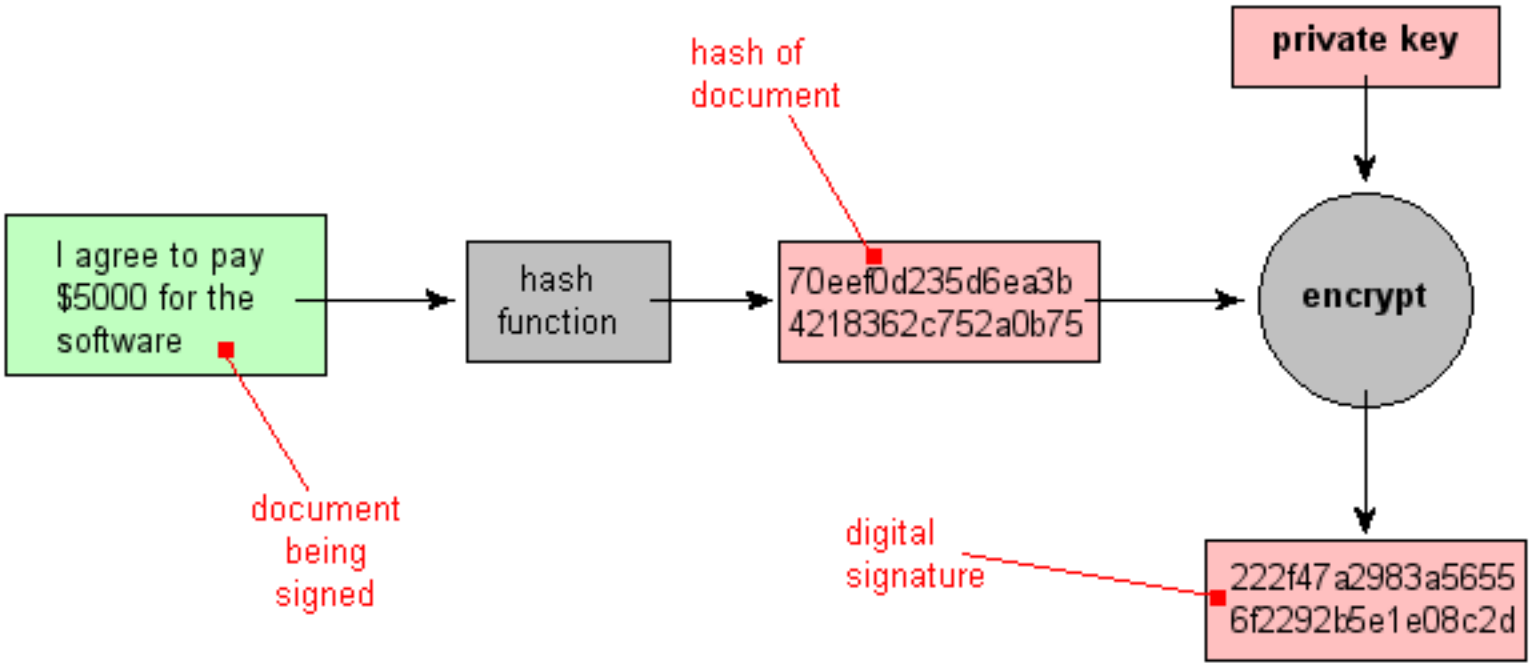


Zauważmy, że aby ta metoda mogła działać, algorytm szyfrowania asymetrycznego musi równie dobrze być w stanie szyfrować kluczem prywatnym i deszyfrować kluczem publicznym, jak i na odwrót. Przykładem algorytmu o tej własności jest RSA.

Skróty kryptograficzne w podpisach cyfrowych

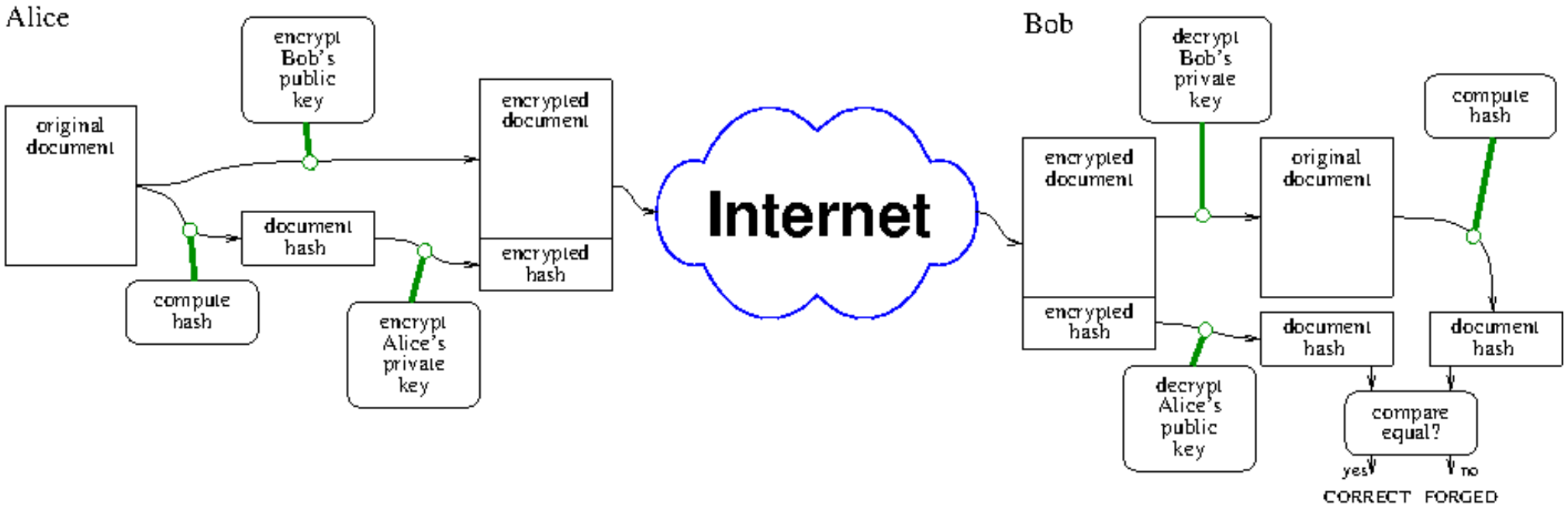
Idea dodania do wysyłanego komunikatu jego zaszyfrowanej wersji jako podpisu działa, ale niekoniecznie jest to wygodne. Na przykład, dla długich komunikatów (takich jak film wideo) podpis byłby niepotrzebnie długi, oraz długo trwałoby jego szyfrowanie.

Zamiast szyfrować całego dokumentu, wystarczy jako podpis cyfrowy zaszyfrować jego skrót kryptograficzny, który jest krótki i z definicji prawie jednoznacznie identyfikuje dokument.



Szyfrowanie i podpisywanie

Zastosowanie skrótów kryptograficznych jest zatem standardową i wygodną metodą podpisywania cyfrowego dokumentów. Poniżej przedstawiona jest pełna procedura szyfrowania dokumentu do bezpiecznej transmisji przez sieć, oraz generowania podpisu cyfrowego w celu sprawdzenia integralności dokumentu i wiarygodności jego autorstwa:



Dokładna zgodność obliczonego przez odbiorcę skrótu z wersją rozszyfrowaną z dokumentu świadczy o zgodności dokumentu z wersją wysłaną przez nadawcę, i jednocześnie potwierdza tożsamość nadawcy.

System klucza publicznego

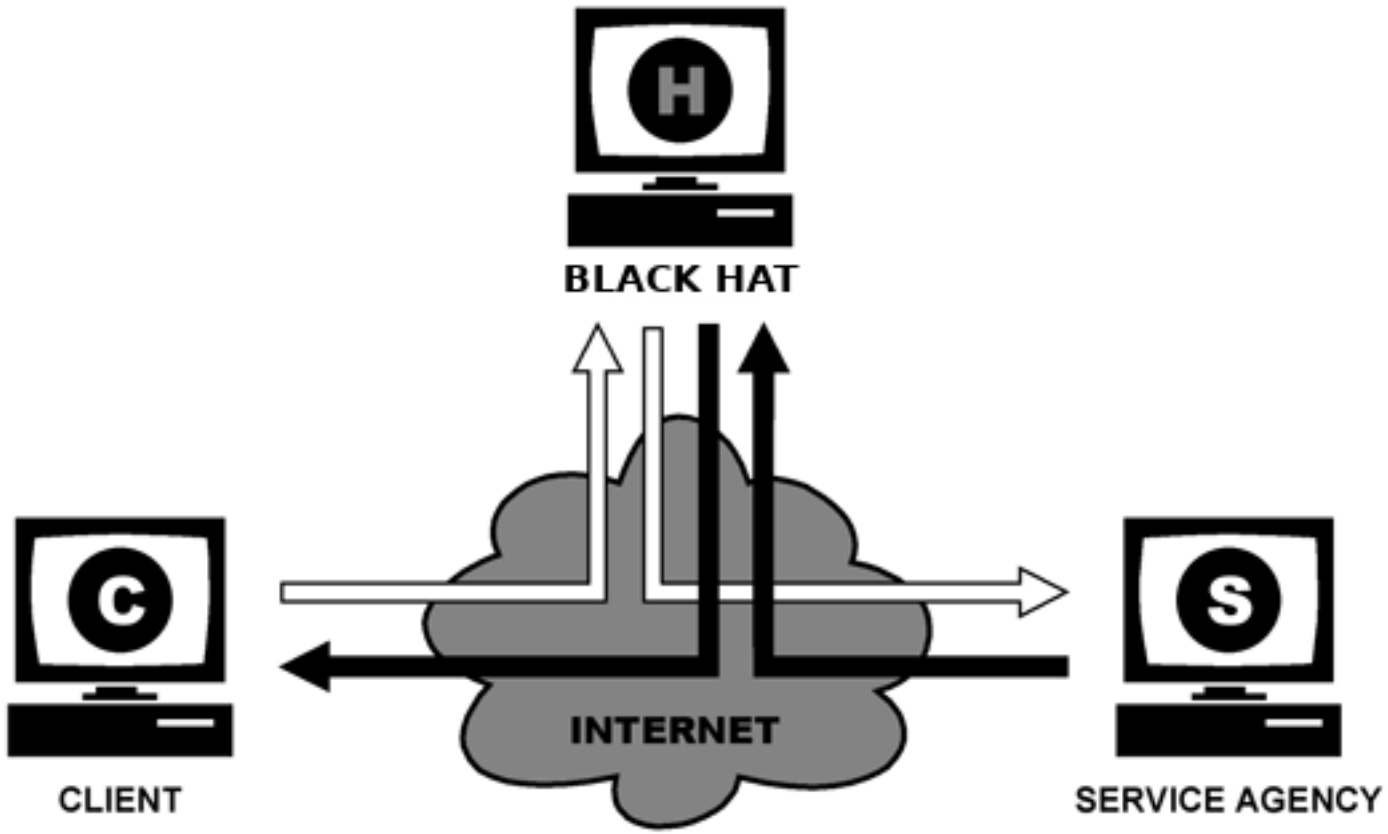
Podsumujmy wiadomości o systemie kluczy publicznych. Pozwala on na zaszyfrowanie komunikatu kluczem publicznym odbiorcy, i jednocześnie wygenerowanie podpisu cyfrowego dokumentu, czyli skrótu kryptograficznego oryginalnego dokumentu zaszyfrowanego kluczem prywatnym nadawcy. Odbiorca może odszyfrować wiadomość swoim kluczem prywatnym, następnie obliczyć jej skrót, i porównać go z otrzymanym od nadawcy skrótem, rozszyfrowanym kluczem publicznym nadawcy. W ten sposób oryginalna wiadomość była bezpiecznie przesłana w postaci zakodowanej, i odbiorca ma jednocześnie gwarancję, że odebrana wiadomość jest dokładnie zgodna z wersją nadaną.

Teoretycznie technologia klucza publicznego rozwiązuje problem dystrybucji kluczy szyfrowania. Klucze można przesyłać jawnie otwartymi kanałami. Każdy może np. opublikować swój klucz na stronie internetowej, albo rozsyłać go elektronicznie bez obawy naruszenia poufności danych.

Jednak w masowym użyciu, z jakim mamy do czynienia we współczesnym Internecie, pojawiają się dodatkowe problemy. Klucze ulegają utraceniu i muszą być sprawnie unieważniane i rozsyłane nowe. Niezawodnie można przesłać klucz publiczny przyjacielowi (lub przyjaciółce), ale jak upewnić się, że klucz publiczny banku, firmy Paypal, albo urzędu skarbowego nie został przekłamany? I co, gdyby tak się stało?

Atak pośrednika

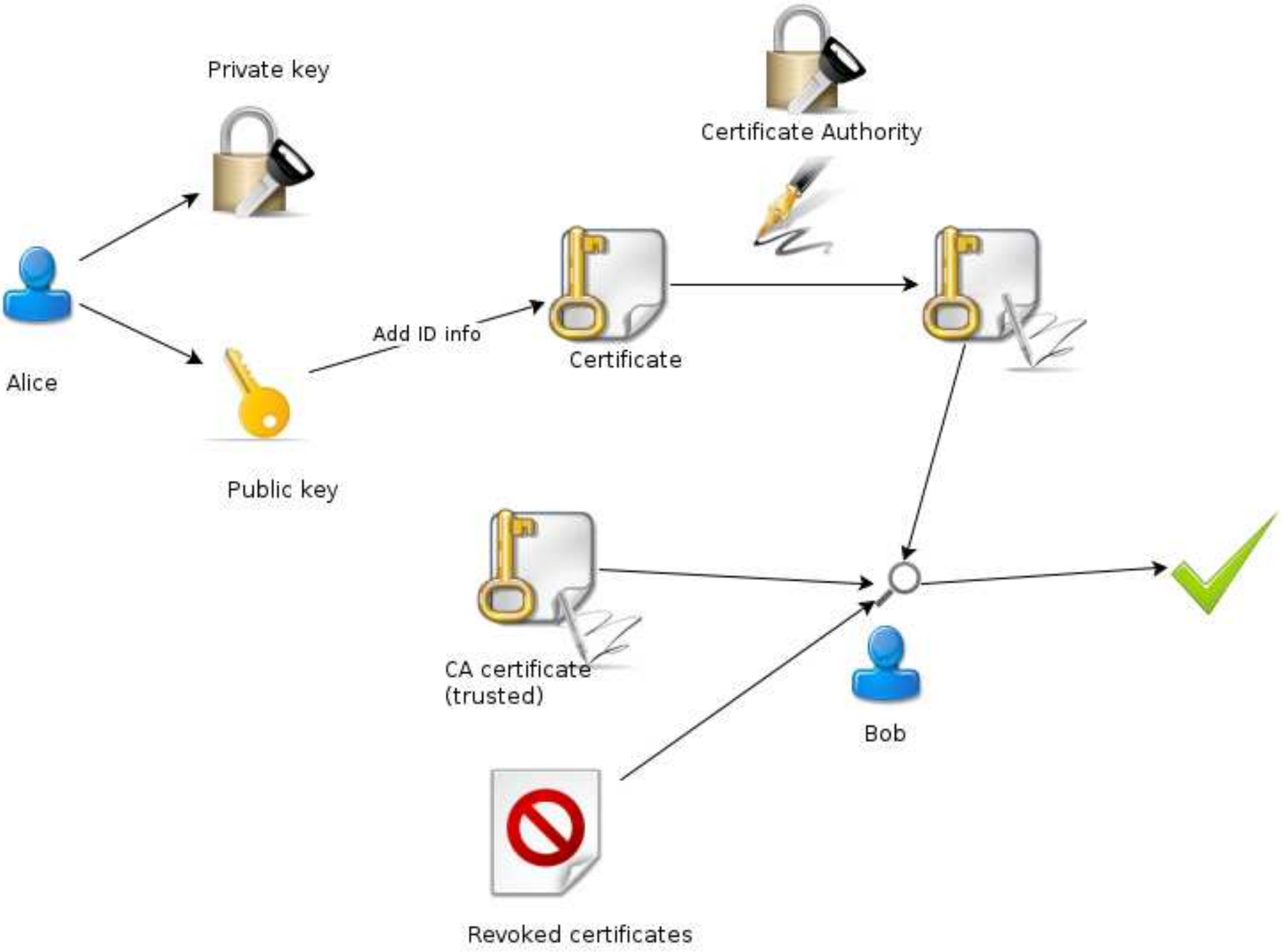
Niestety, w Internecie rozwinęły się liczne techniki ataków wykorzystujące dziury w zabezpieczeniach. Jeżeli komuś uda się przeprowadzić atak w chwili pobierania kluczy publicznych do komunikacji między dwoma partnerami, to może łatwo przechwycić, a nawet sfałszować, całą komunikację między nimi wykorzystując schemat zwany **atakiem pośrednika** (ang. *man-in-the-middle attack*).



Infrastruktura klucza publicznego (PKI)

Z powyższych względów system klucza publicznego został w Internecie rozbudowany do **Infrastruktury Klucza Publicznego** (PKI — *Public Key Infrastructure*). Wymaga ona stworzenia zaufanej instytucji zwanej Centrum (albo Urzędem) Certyfikacji CA (*Certification Authority*). Jego rolą jest generowanie **certyfikatów** potwierdzających, że dany klucz publiczny jest rzeczywiście kluczem osoby lub instytucji, która podaje się za właściciela klucza. Ponieważ certyfikat jest podpisany przez CA, więc każdy może sprawdzić, że przesłany klucz publiczny innej jednostki jest właściwy.

Jednocześnie CA przechowuje informacje o unieważnionych kluczach, i pełni szereg dalszych funkcji przydatnych w procesie szyfrowania i podpisywania dokumentów między partnerami w Internecie. Z tego powodu Centrum Certyfikacji musi samo być jednostką w pełni zaufaną, którego ani tożsamość, ani wiarygodność kluczy szyfrowania nie budzą wątpliwości. Na przykład, przeglądarki internetowe mają wbudowane w sobie listy istniejących na świecie CA, i normalnie nie przyjmują certyfikatów podpisanych przez CA spoza tej listy.


Private key
Certificate Authority
Alice
Add ID info
Certificate
Public key
CA certificate
(trusted)
Bob
Revoked certificates

Uwierzytelnianie

Użytkownicy systemów komputerowych mają przyznane różne uprawnienia. Aby system udostępnił użytkownikowi dostęp do zasobów, do których ma uprawnienia, musi on/ona: (a) zidentyfikować się, to znaczy podać swoją tożsamość, i (b) uwierzytelnić się, to znaczy, potwierdzić tę tożsamość.

Problem identyfikacji i uwierzytelniania istnieje w wielu systemach, nie tylko komputerowych. Taka potrzeba zachodzi np. przy wypożyczaniu książki z biblioteki, wyłączania systemu alarmowego, albo wypłaty gotówki z bankomatu. Nie jest to również potrzeba związana ze współczesnymi czasami ani nowymi technologiami. Już w starożytności wysłannik jednego władcy, aby być jego uznanym reprezentantem w siedzibie innego, musiał się uwierzytelnić.

Tradycyjnie, istnieją trzy podstawowe sposoby uwierzytelniania:

- za pomocą czegoś, co użytkownik wie,
- za pomocą czegoś, co użytkownik posiada,
- za pomocą czegoś, co użytkownika charakteryzuje.

W praktyce te trzy sposoby z reguły przekładają się użycie: haseł, tokenów (żetonów?) dostępu, i parametrów biometrycznych.

Uwierzytelnianie z użyciem haseł

Najstarszą powszechnie stosowaną techniką uwierzytelniania użytkowników przy dostępie do zasobów komputerowych były i są hasła. Prawidłowo stosowana, ta metoda może nadal być skuteczna, jednak masowe stosowanie haseł do wszystkiego powoduje, że prawie nie ma możliwości opanować „prawidłowej" techniki stosowania haseł.

W niektórych wczesnych systemach hasła przechowywane były w wersji źródłowej, co w oczywisty sposób tworzy zagrożenie w przypadku wycieku takiej bazy danych haseł. System Unix wprowadził metodę przechowywania haseł w formie zahaszowanej funkcją jednokierunkową, z jednoczesną zasadą dostępności wszystkich zahaszowanych haseł.

Niestety, ta metoda wymyślona i skuteczna w latach 1970-tych, przestała być bezpieczna w dobie masowych ataków i szybkich komputerów. Jest ona podatna na tzw. **ataki słownikowe**. Krakerzy budują listy potencjalnych haseł złożone ze wszystkich słów danego języka, oraz wszystkich nazw własnych: imion, popularnych nazwisk, nazw geograficznych, itp. Zarówno takie pojedyncze słowa, jak i ich popularne wariacje są haszowane znanym systemowym algorytmem, i kraker przygotowuje sobie listę zahaszowanych haseł. W przypadku przejęcia listy haseł z jakiegoś systemu, w krótkim czasie można porównać je wszystkie z przygotowanymi hasłami, wykrywając potencjalnie wiele kont, które włamywacz może następnie wykorzystać do ataku na system komputerowy, jako stację przesiadkową do kolejnego ataku, itp.

Bezpieczeństwo haseł — z punktu widzenia użytkownika

siła hasła — W oczywisty sposób trywialne hasła (basia123) stanowią słabe zabezpieczenie i są podatne na złamanie. Jednak trudne do złamania hasła stanowią utrudnienie w codziennym użyciu, a przy wielu hasłach którymi użytkownik musi posługiwać się na co dzień, zmusiłyby go/ją do zapisywania sobie haseł w miejscu łatwo dostępnym (np. na żółtej karteczce przyklejonej do monitora). Jest to zatem kompromis — użytkownik musi rozważyć jaka jest uciążliwość stosowania bezpiecznych haseł w porównaniu z zagrożeniem (utrata zawartości komputera, pieniędzy, pracy, proces sądowy, kompromitacja zawodowa lub rodzinna, itp.).

bezpieczeństwo wpisywania — Hasło może zostać przechwycone w momencie jego wpisywania przez użytkownika. Na przykład: fizycznie na ekranie, jeśli jest wyświetlane. Albo przez specjalne urządzenia przechwytujące (tzw. *keyloggery*) wpięte fizycznie w kabel klawiatury, lub program zainstalowany na komputerze. Albo z kolei przez nasłuchiwanie transmisji sieciowej (kablowej lub WiFi), gdy użytkownik wpisuje hasło do zdalnego systemu, i jest ono przesłane otwartym tekstem przez sieć.

fizyczny dostęp do komputera — Hasło do konta ustawione na komputerze, do którego włamywacz uzyska dostęp fizyczny (bo dostanie się do pomieszczenia, lub ukradnie komputer), w efekcie będzie nieefektywne, o ile setup komputera nie został zabezpieczony hasłem, i/lub dysk nie został zaszyfrowany.

Bezpieczeństwo haseł — z punktu widzenia programisty/administratora

przechowywanie haseł — Rolą programistów i/lub administratorów systemów komputerowych jest bezpieczne przechowywanie haseł. W praktyce oznacza to przechowywanie tylko wersji zahaszowanej celowo mało efektywnym obliczeniowo algorytmem, z jednoczesnym zabezpieczeniem dostępu do listy takich haseł. Powinny one być dostępne tylko za pośrednictwem uprzywilejowanego programu, który pobiera pojedyncze zahaszowane hasło w celu jego porównania z wersją uzyskaną od użytkownika.

utrudnienia dla włamywaczy — Warto i należy stosować proste techniki utrudniające skanowanie kont i haseł: kilkusekundowe opóźnienia zapytania o konto i hasło skutecznie wyklucza większość programów skanujących, a blokowanie adresów skąd przychodzą takie próby skanowania zabezpiecza system przez ponawianymi atakami.

sól/pieprz — Technika usprawniająca haszowanie haseł — każde hasło przed zahaszowaniem jest uzupełniane losowo wygenerowanym ciągiem bitów zwanym **solą** (*salt*). Sól jest pamiętana jawnie razem z zahaszowanym hasłem, aby przy sprawdzaniu hasła podanego przez użytkownika prawidłowo je zahaszować z solą. Jednak krakerzy nie mogą korzystać z gotowej listy haseł słownikowych. Dla każdego znalezionego hasła użytkownika lista słownikowa musi być indywidualnie budowana i haszowana. **Pieprz** (*pepper*) działa podobnie jak sól, tylko nie jest jawnie przechowywany razem z zahaszowanym hasłem.

Dygresja — ćwiczenia z haszowaniem haseł z solą

Haszowanie hasła domyślną metodą DES, hasz ma 13 znaków, dwa pierwsze to sól:

```
whistler-572> mkpasswd -m des basia123     # losowa wartosc soli
afRI7OBQ18nIc
whistler-573> mkpasswd -m des basia123     # losowa wartosc soli
UT/tZsoSy6ZVs
whistler-573> mkpasswd -m des basia123     # losowa wartosc soli
a1B4DTn8jirTs
whistler-573> mkpasswd -m des basia123 a1  # wymuszona ta sama wartosc soli
a1B4DTn8jirTs
```

Haszowanie hasła metodą MD5 (id1), hasz ma 22 znaki, sól 8 znaków:

```
whistler-592> mkpasswd -m md5crypt basia123
$1$CzEaBRPm$rbhG9s2Zm6UCvH3kmxELV0
whistler-593> mkpasswd -m md5crypt basia123
$1$aBGCR155$B9QwheDcsEA1DIza/bvDl0
whistler-593> mkpasswd -m md5crypt basia123
$1$R6O2m1DL$2WvgqQ870EU1RS9kzTWPv0
whistler-594> mkpasswd -m md5crypt basia123 R6O2m1DL
$1$R6O2m1DL$2WvgqQ870EU1RS9kzTWPv0
```

MD5-Crypt haszuje metodą MD5 1000 razy stringa: hasła+soli+poprzedniego haszu.

Bezpieczeństwo haseł — z punktu widzenia producenta/instalatora systemów

domyślne hasła do urządzeń — Wiele różnych produkowanych na świecie urządzeń, jak również systemów oprogramowania, dostarczanych jest z preinstalowanymi fabrycznymi hasłami, których wielu użytkowników nigdy nie zmienia. Otwiera to ogromne pole do działania czarnych kapeluszy.

Ma to szczególne znaczenie zwłaszcza w połączeniu z modnymi technologiami IoT (*Internet of Things*) instalującymi procesory z dostępem do internetu na wielu urządzeniach codziennego użytku. Urządzenia te mają powszechnie znane domyślne hasła, bardzo słaby ogólny poziom zabezpieczeń, i liczne dziury, również ogólnie znane. Niektóre zamontowane są w urządzeniach, które mogą być niebezpieczne.

Na przykład przeprowadzone w roku 2010 ataki za pomocą robaka Stuxnet były możliwe dzięki wykorzystaniu domyślnego hasła do przemysłowego oprogramowania bazy danych SCADA firmy Siemens. Wskutek tych ataków ucierpiał m.in. irański zakład wzbogacania uranu, gdzie uszkodzeniu uległy wirówki przyspieszone poza krytyczną prędkość obrotową przez oprogramowanie sterujące.

Uwierzytelnianie z żetonem

Uwierzytelnianie z żetonem nie jest nowym wynalazkiem. Od wieków stosowane były różne formy żetonów uwierzytelniania: wymyślny sygnet potwierdzający tożsamość osoby, pieczęć królewska uwierzytelniająca dokumenty, albo klucz umożliwiający dostęp do kufra z zasobami (lub do wieży z zamkniętą księżniczką).

Współczesne wersje tokenów uwierzytelniania (żetonów) to: karty bankowe, czyli karty pamięci z hasłem zapamiętanym w postaci magnetycznej lub w elektronicznej pamięci ROM, tokeny generujące hasła jednorazowe, albo znacznie bardziej zaawansowane inteligentne karty procesorowe.

Pewną wersją uwierzytelniania z żetonem jest technika haseł jednorazowych. Użytkownik posiada listę takich haseł pierwotnie wygenerowanych na serwerze, i wykorzystuje je po kolei, każde tylko jeden raz. W praktyce, takie hasła działają bardziej jak żeton uwierzytelniania niż hasło, i ta metoda eliminuje wiele wad tradycyjnego systemu haseł.

Alternatywą dla listy jednorazowo wygenerowanych haseł jednorazowych są elektroniczne tokeny generujące takie hasła zsynchronizowane z aktualnym czasem.

Uwierzytelnianie z inteligentną kartą procesorową

O ile karty pamięci, tak magnetyczne jak i elektroniczne, są podatne na możliwość skopiowania, to tej wady nie mają karty inteligentne, zawierające procesor i niewielką pamięć. Uwierzytelnianie z użyciem takiej karty przebiega w trybie **zapytanie-odpowiedź** (*challenge-response*), gdzie karta nie odpowiada na zapytanie hasłem, tylko używa go do zaszyfrowania i deszyfrowania na wewnętrznym procesorze.

Karty inteligentne zawierające procesor i niewielką pamięć w połączeniu z metodą zapytanie-odpowiedź zapewniają bardzo wysoki poziom bezpieczeństwa. Hasło po pierwotnym wygenerowaniu jest zapisywane w pamięci karty i nie jest nigdy nigdzie przesyłane. Uwierzytelnianie polega na wygenerowaniu losowego tekstu i przesłaniu go do procesora karty, a następnie uzyskaniu od karty wersji tego tekstu zaszyfrowanej kluczem prywatnym karty. Jest on następnie rozszyfrowywany pamiętanym na serwerze kluczem publicznym, i porównaniu z wersją oryginalną tekstu. Ponieważ tekst jest jednorazowy, to ani znajomość jego postaci oryginalnej, ani zaszyfrowanej, nie daje żadnej możliwości przyszłego ataku.

Podobne technologie można zastosować z wykorzystaniem smartfonów, jednak smartfony same nie są systemami bezpiecznymi. Wręcz przeciwnie, same są wdzięcznym obiektem do różnych metod ataku.

Uwierzytelnianie biometryczne

Uwierzytelnianie biometryczne jest najstarszą techniką uwierzytelniania. Rozpoznawanie twarzy, sylwetki, i innych cech biometrycznych było stosowane przy wpuszczaniu naszych przodków do jaskini zamieszkałej przez dane plemię. Delfiny (i wiele innych gatunków zwierząt) stosują sygnatury dźwiękowe do wzajemnego rozpoznawania. Koty znaczą terytorium indywidualnym zapachem.

Technologie biometryczne takie jak: rozpoznawanie odcisku palca, obrazu twarzy, obrazu siatkówki oka, głosu, itp., są coraz częściej stosowane w systemach dostępowych (i komputerowych). Są one wygodniejsze od haseł, a nawet od kart inteligentnych, i w teorii zapewniają wysoki poziom bezpieczeństwa. Jednak wiele z nich jest jeszcze w fazie intensywnego rozwoju i w praktyce nie są jeszcze aż tak wygodne i niezawodne jak te stosowane przez naszych przodków.

Niektóre wady uwierzytelniania biometrycznego:

- system ustawiony na wysoki wymagany poziom zgodności może nie rozpoznać użytkownika w stanie zmęczenia, makijażu, użycia soczewek kontaktowych, itp.; natomiast ustawiony na niski wymagany poziom zgodności może uznawać podobne obrazy za zgodne
- system może budzić zastrzeżenia etyczne, gdy będzie różnie traktował użytkowników różnej płci, wieku, rasy, budowy fizycznej, itp.
- bezpieczne przechowywanie danych biometrycznych jest wysoce krytyczne; w przypadku ich wykradzenia nie ma możliwości zresetowania danych jak haseł