

Kryptografia i mechanizmy bezpieczeństwa

Paweł Krawczyk

Kontakt z autorem:

pawel.krawczyk@hush.com

Tel. +48-602-776959

Prezentacja udostępniona na licencji Creative Commons BY-NC-SA
(„uznanie autorstwa”, „użycie niekomercyjne”, „na tych samych warunkach”)

<http://creativecommons.org/licenses/by-nc-sa/3.0/pl/>

Konspekt

- Funkcje kryptografii
- Podstawowe operacje kryptograficzne
- Ataki kryptoanalityczne
- Podstawowe techniki kryptograficzne
- Zarządzanie kluczami

Funkcje kryptografii

- Poufność
 - Szyfrowanie
- Integralność
 - Funkcje skrótu
- Rozliczalność
 - Protokoły uwierzytelniania, integralność logów
- Anonimowość
 - Dowód z wiedzą zerową, szyfrowanie
- Dostępność
 - Kontrola dostępu

Terminologia

- Tekst jawny, otwarty (*plain, clear text* - P)
- Szyfrogram, kryptogram (*cipher text* – C)
- Szyfrowanie – deszyfrowanie (rozszyfrowanie)
- Klucz (*key*)
 - Tajny parametr – realizuje kontrolę dostępu do tekstu jawnego
- Atak kryptoanalityczny
 - Teoretyczny
 - Praktyczny

Terminologia

- Algorytm kryptograficzny
 - Funkcja bezpieczeństwa
 - DES, AES, MD5, SHA-1, RSA...
- Protokół kryptograficzny
 - Operacja
 - ESP (IPSec), SSL Record Protocol (TLS)
- System kryptograficzny (*cryptosystem*)
 - Funkcja biznesowa
 - PGP, X.509

Zasada Kerckhoffsza

- Zasada Kerckhoffsza
 - Auguste Kerckhoffs 1883
 - Bezpieczeństwo systemu powinno polegać na tajności klucza, a nie systemu
 - Claude Shannon
 - „Wróg zna szczegóły systemu”

Jawność algorytmów kryptograficznych

- Algorytmy jawne
 - Niezależna weryfikacja poprawności
 - Cryptology ePrint Archive (IACR), ArXiv.org
 - Większość algorytmów cywilnych jest jawna
- Algorytmy niejawne
 - Dodatkowe zabezpieczenie, utrudnia analizę
 - Tylko jeśli skuteczne ograniczenie dostępu do implementacji jest możliwe
 - Nie dotyczy większości implementacji komercyjnych

Standardy kryptograficzne

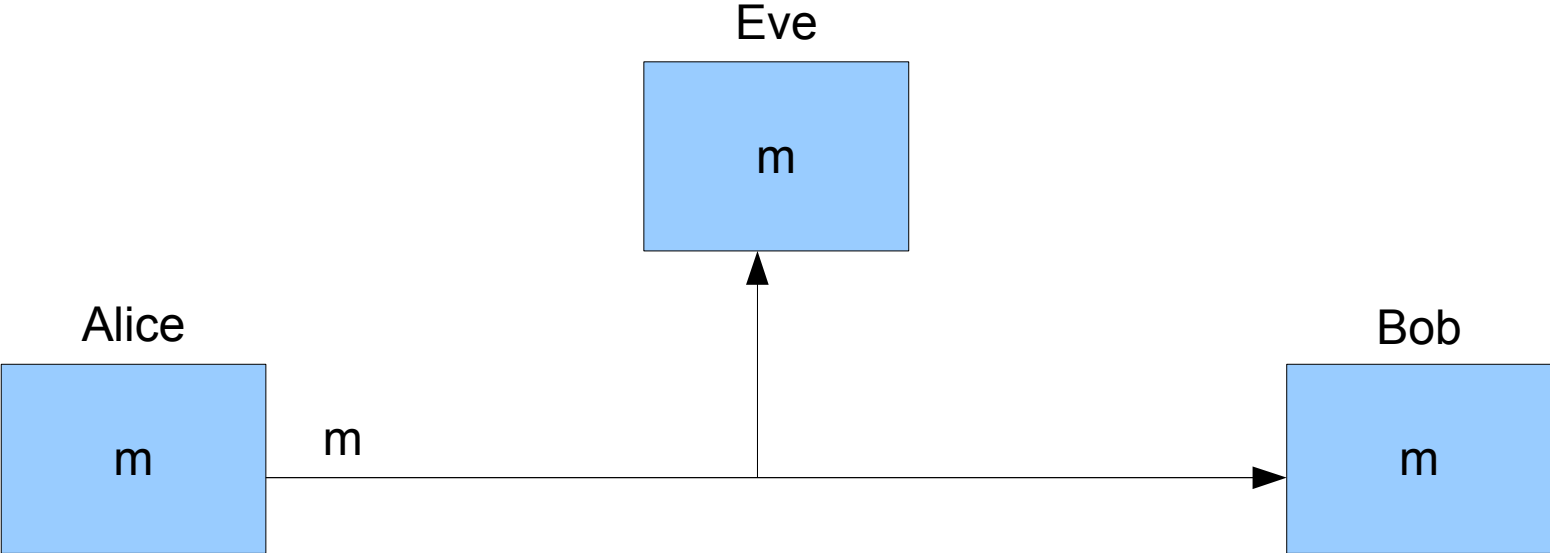
- Odpowiedzialność projektanta systemów
- Wybór technik kryptograficznych
 - 100% oparcia w standardach i normach
 - Regulacje polskie i europejskie (uodo, uope, uooin, bankowe)
 - NIST Computer Security Resources Center
 - NIST Federal Information Processing Standards (FIPS)
 - NIST Special Publications (zalecenia)

Standardy kryptograficzne

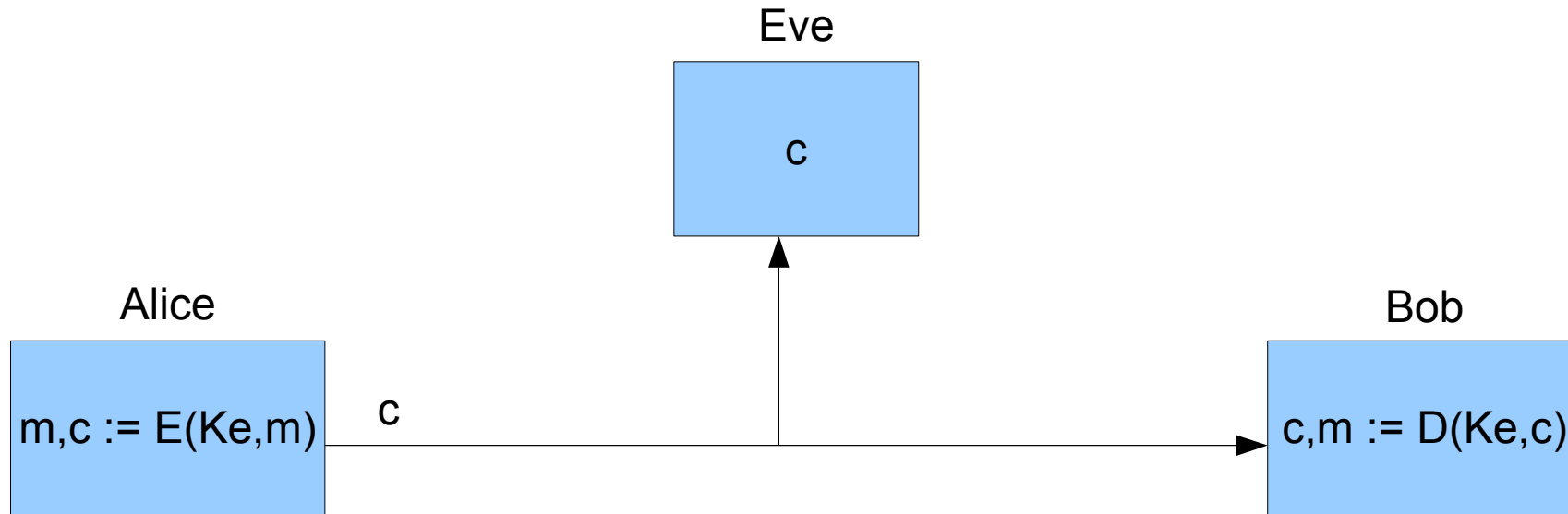
- Wybór implementacji kryptograficznych
 - Wbudowane w system lub framework aplikacji
 - Microsoft Cryptographic API (CAPI)
 - Biblioteki kryptograficzne
 - Komercyjne, otwarte
 - Certyfikaty bezpieczeństwa

Podstawowe operacje kryptograficzne

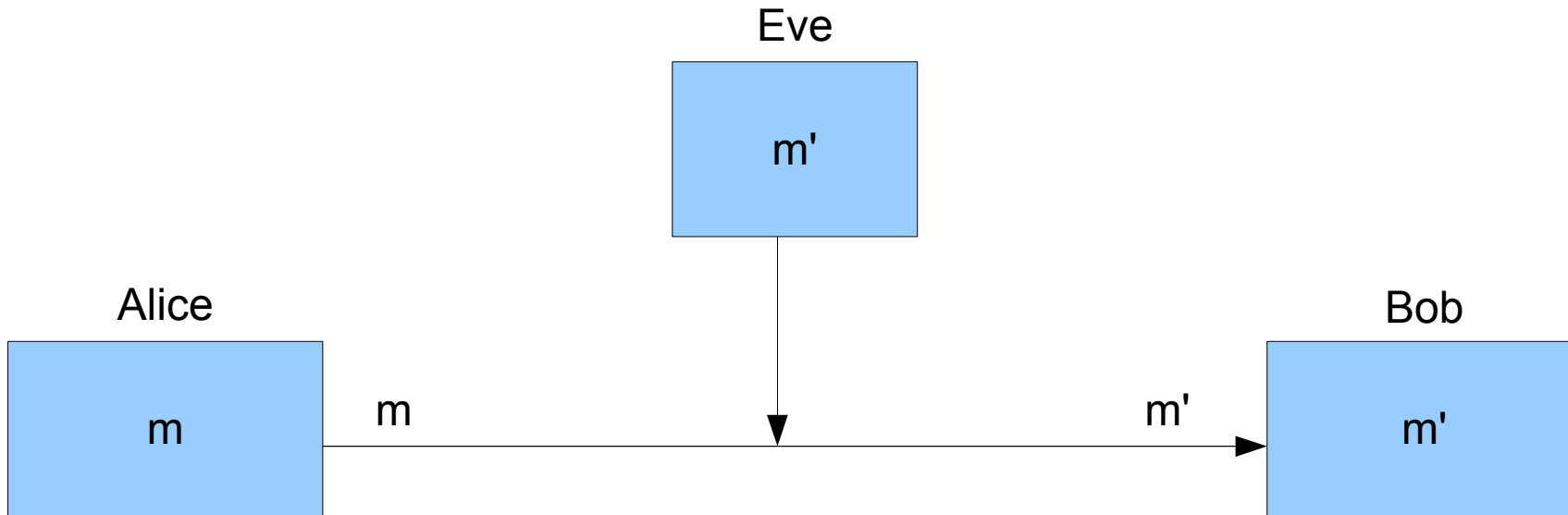
Atak: podsłuch



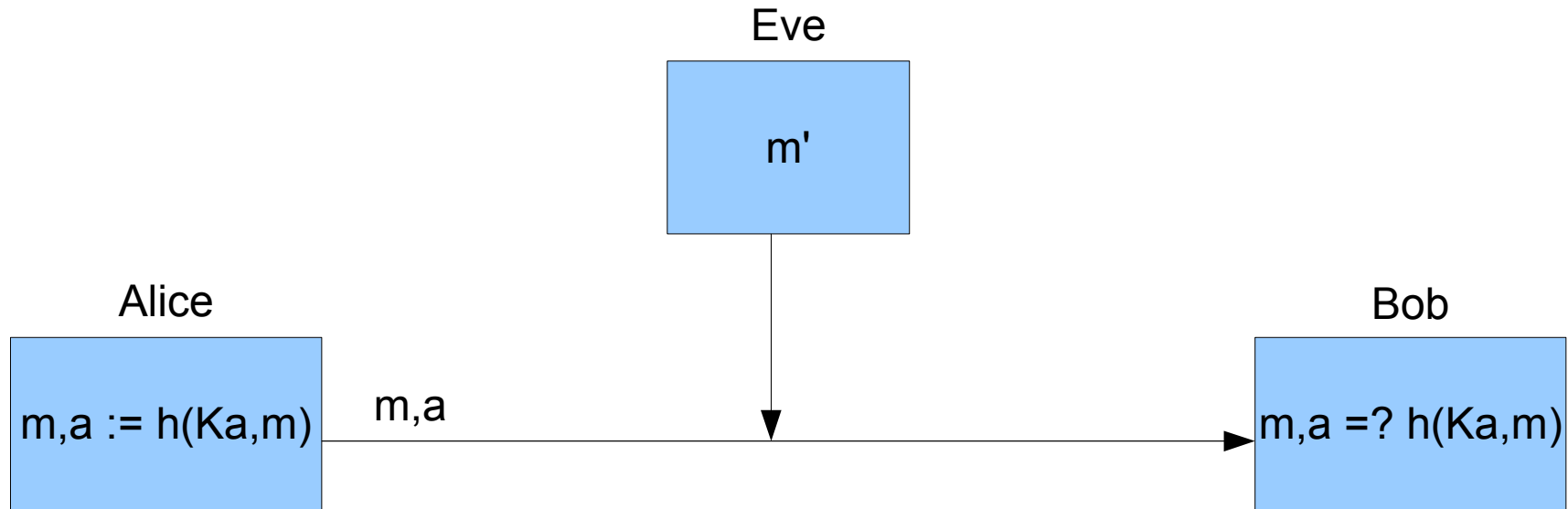
Obrona: szyfrowanie



Atak: fałszerstwo



Obrona: uwierzytelnienie



Koncepcja klucza publicznego

- Kryptografia symetryczna
 - Alice i Bob stosują ten sam klucz (K_e , K_a)
 - Źle skalowalny (10 os. - 45 kluczy, 20 – 190 itd)
- Kryptografia asymetryczna
 - Alice ma teraz
 - Klucz do szyfrowania (publiczny – jawny)
 - P_{Alice} (*public*)
 - Klucz do deszyfrowania (prywatny – tajny)
 - S_{Alice} (*secret*)
- Public Key (PK) Cryptography (PKC)

Szyfrowanie kluczem publicznym



Jeszcze o kluczach publicznych

- Wiadomość zaszyfrowana P_x
 - Tylko posiadacz S_x może ją odczytać
 - Nawet nadawca jej nie odczyta
 - Jeśli np. zgubi oryginał
 - W praktyce – nadawca używa i P_x i swojego P
- Znacząco zredukowana liczba kluczy
 - Każdy publikuje tylko swój klucz

Szyfrowanie hybrydowe

- Wady PKC
 - Wymagające obliczeniowo i pamięciowo
 - O rząd wielkości wolniejsze niż szyfry symetryczne
- Rozwiązanie
 - Wiadomość m szyfrujemy Ke
 - Duża porcja danych, szybki szyfr
 - Ke szyfrujemy Px
 - Mała porcja danych, wolny szyfr
 - Odbiorca deszyfruje Ke swoim Sx
 - Używając Ke deszyfruje m

Podpis cyfrowy



Terminologia e-podpisowa

- Pojęcia techniczne i prawne – kolizje
- Podpis cyfrowy
 - Pojęcie inżynierskie
 - Poświadczenie autentyczności i integralności danych
 - Różne zastosowania
- Podpis elektroniczny
 - Pojęcie prawno-inżynierskie
 - Związany z osobą fizyczną

Autentyczność kluczy

- Klucz K_a lub P_x zapewnia autentyczność danych
 - Ale co zapewnia autentyczność K_a i P_x ?
 - Konieczne rozwiązania organizacyjne
- Poświadczenie kluczy
 - Wzajemne (mesh)
 - Zaufana trzecia strona (TTP – *Trusted Third Party*)
 - Urzędy certyfikacji, centra certyfikacji (CA – *Certifying authority*)
 - Informacja o aktualności kluczy

PKI

- Infrastruktura klucza publicznego
 - PKI – *Public Key Infrastructure*
 - Środki techniczne, organizacyjne i prawne
 - Systemy kryptograficzne
 - TTP świadczące usługi certyfikacyjne
 - Osadzenie w prawie

Ataki kryptoanalityczne

Cel ataku na szyfr

- Cel szyfru
 - Utrzymanie złożoności ataku na poziomie przeszukiwania wszystkich kluczy (*brute force, exhaustive*)
- Pośredni cel ataku
 - Obniżenie złożoności poniżej tej wartości
- Bezpośredni cel ataku
 - Uzyskanie dostępu do wiadomości

Obiekt ataku

- Ataki na algorytmy
 - Słabości w algorytmach – Enigma, Lorenz, PURPLE, RC4
- Ataki na implementacje
 - WEP (RC4), PPTPv1 (RC4), OpenSSL (RSA)
- Ataki na użytkownika
 - Kleptografia, "*rubber-hose cryptanalysis*"

Wiedza atakującego

- Tylko kryptogram (*ciphertext only*)
 - Zawsze
- Znany tekst jawny (*known-plaintext*)
 - Przewidywalny tekst jawny, tekst podestłany
- Wybrany tekst jawny (*chosen plaintext*)
- Wybrany tekst jawny i zaszyfrowany (*chosen ciphertext*)

Przykłady ataków

- Paradoks dnia urodzin (*birthday paradox*)
 - W grupie 23 osoby prawdopodobieństwo urodzin tego samego dnia przekracza 50% ("kolizja")
 - Wśród identyfikatorów transakcji z przedziału 2^{64} – kolizje po 2^{32} transakcjach
 - W zbiorze N elementów – 50% kolizji po \sqrt{N}
 - Duże znaczenie dla
 - Identyfikatorów wiadomości (*anti-replay*)
 - Wartości, które nie mogą się powtarzać (niektóre klucze)

Przykłady ataków

- Spotkanie w środku (*meet-in-the-middle, collision attack*)
 - Pregenerowanie "słownika"
 - 2^{32} par: tekst jawny \rightarrow kryptogram
 - Śledzenie komunikacji
 - Kolizja prawdopodobna już po 2^{32} kryptogramach
 - Złożoność po ataku: $2^{32} + 2^{32} = 8,6 \times 10^9$
 - Oryginalna złożoność: $2^{64} = 1,8 \times 10^{19}$

Realna siła algorytmów

- Idealnie – równa długości klucza
 - Klucz 2^{128} bitów $\rightarrow 2^{128}$ kroków do złamania klucza
- Znane ataki mogą zmniejszać złożoność
 - 2TDES – klucz 112 bitów – siła ok. 80 bitów
 - 3TDES – klucz 168 bitów – siła ok. 112 bitów
 - DES-X – klucz 184 bity – siła ok. 118 bitów

Podstawowe techniki kryptograficzne

Szyfr blokowy

- Pracuje na porcjach danych o stałej długości
 - Bloki - obecnie min. 128 bitów (16 bajtów)
- Symetryczny
 - Ten sam klucz do szyfrowania i deszyfrowania
 - Obecnie min. 128 bitów (16 bajtów)
- Stałe przekształcenie
 - Ten sam algorytm dla tego samego K_e i tego samego P da ten sam C
- Standardowy interfejs
 - Ustaw klucz, szyfruj, deszyfruj

Wymagania wobec szyfrów

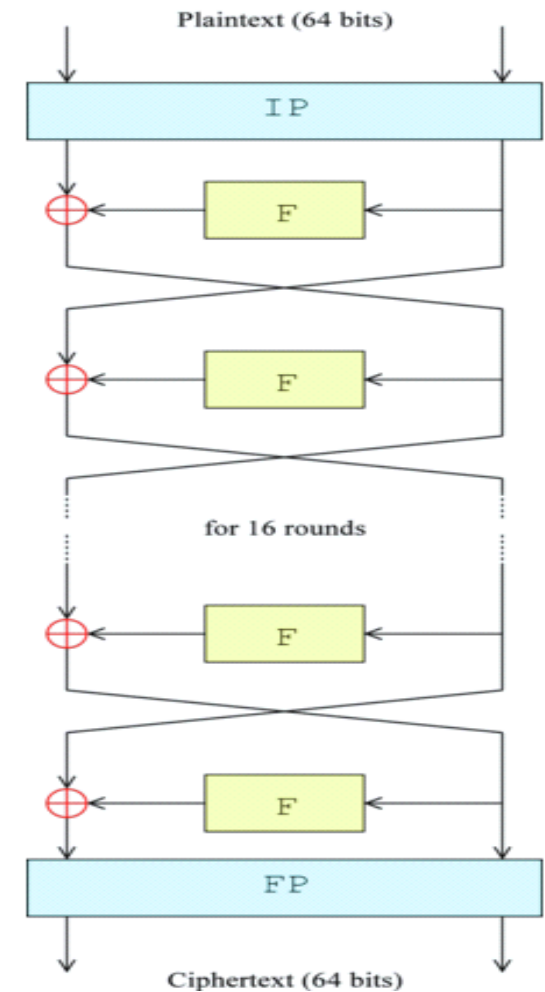
- Efekt lawinowy (*avalanche effect*)
 - Zmiana jednego bitu tekstu jawnego lub klucza zmienia wszystkie bity kryptogramu
- Rozproszenie (*diffusion*)
 - Zatarcie charakterystyki tekstu jawnego w kryptogramie
- Przemieszanie (*confusion*)
 - Zatarcie związku między kluczem a kryptogramem

Szyfry blokowe w praktyce

- DES (Data Encryption Standard)
 - Blok 64 bity, klucz 56 bitów
 - Nie używać
- 3DES
 - Blok 64 bity, klucz 168 bitów
- AES (Advanced Encryption Standard)
 - Blok 128 bitów, klucz 128, 192, 256 bitów
 - Aktualnie zalecany standard cywilny
- Serpent, Twofish, MARS, RC6

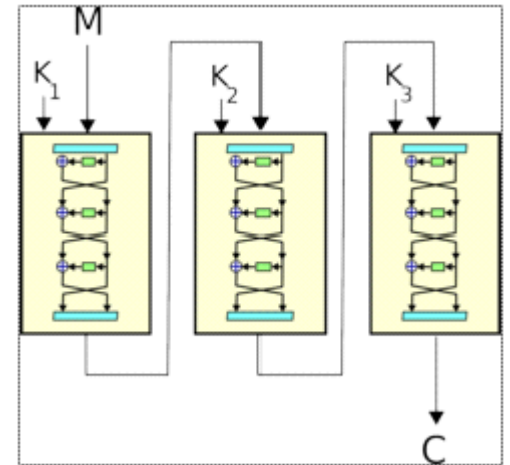
DES

- Standard od 1975
- Niezalecany od 1999 (NIST)
- Brak ataków praktycznych lepszych niż *brute force*
- Wady
 - Krótki klucz (RSA DESCHALL)
 - Krótki blok (kolizje)
 - Niska wydajność



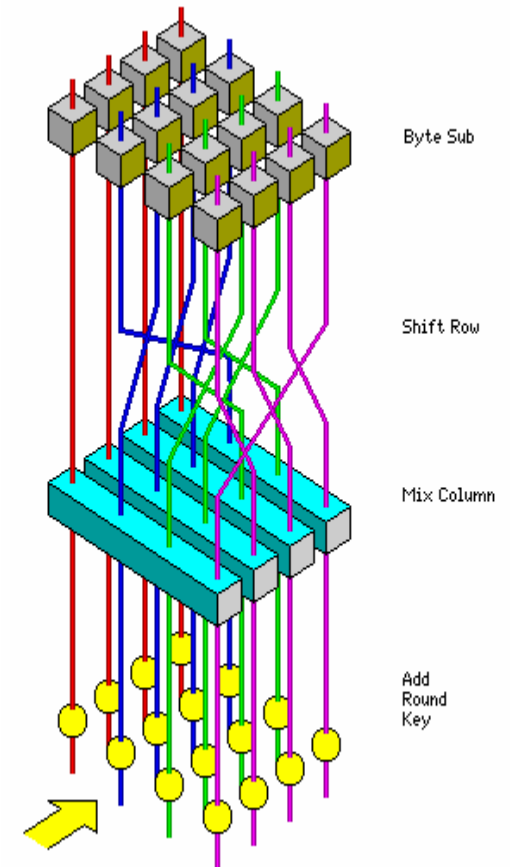
3DES

- Szyfrowanie-deszyfrowanie-szyfrowanie
 - 3DES-EDE
 - Długość klucza $3 \times 56 = 168$ bitów
- Dopuszczalny pod warunkiem używania trzech niezależnych kluczy
- Wady
 - Krótki blok (kolizje)
 - Niska wydajność



AES

- Advanced Encryption Standard (AES)
 - Rijndael
- Bardzo wydajny
- Wg niektórych niski margines bezpieczeństwa



Inne szyfry

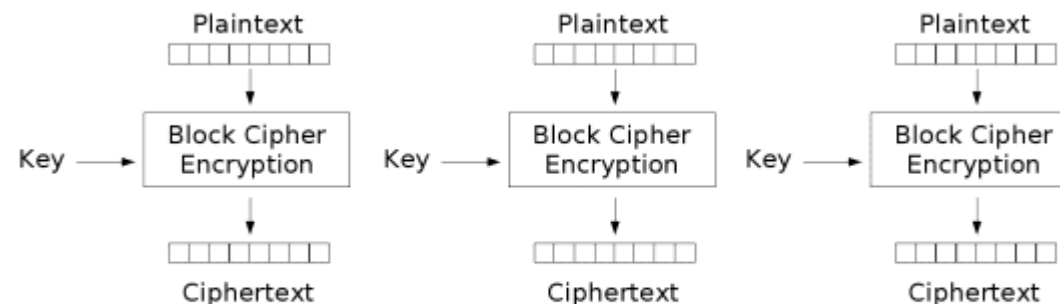
- Serpent
 - Wysoki margines bezpieczeństwa, wolniejszy od AES, darmowy
- Twofish
 - Wydajny, darmowy
- MARS
 - Darmowy
- RC6
 - Opatentowany (RSA)

Tryby szyfrowania

- Blok 128 bitów = 16 bajtów
 - Podział tekstu jawnego na bloki
 - Uzupełnianie bloków (*padding*)
- Różne sposoby podziału na bloki
 - Tryby szyfrowania (*block cipher modes*)
 - Różne istotne konsekwencje dla bezpieczeństwa
 - Dodatkowe parametry
- Standardowy interfejs programistyczny
 - `DES_cbc_encrypt()`, `DES_cbc_decrypt()`

Electronic Code Book (ECB)

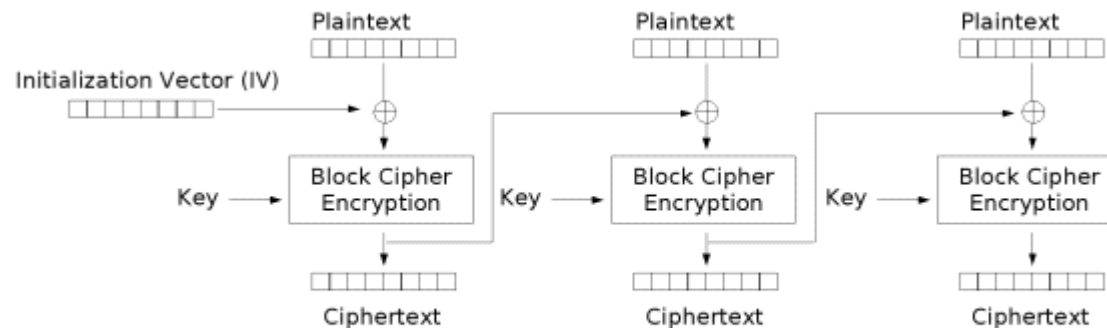
- Najbardziej intuicyjny
 - P dzielony na bloki po 16 bajtów
 - Każdy blok szyfrowany samodzielnie
 - $C_i := E(K, P_i)$, $P_i := D(K, C_i)$
- Taki sam blok P da taki sam blok C
 - Szereg problemów związanych z bezpieczeństwem



Electronic Codebook (ECB) mode encryption

Cipher Block Chaining (CBC)

- Każdy blok wejściowy jest dodawany do poprzedniego kryptogramu
 - Operacja XOR
 - $C_i := E(K, C_{i-1} \oplus P_i)$
 - $P_i := D(K, C_i) \oplus C_{i-1}$
- Najbardziej rozpowszechniony tryb



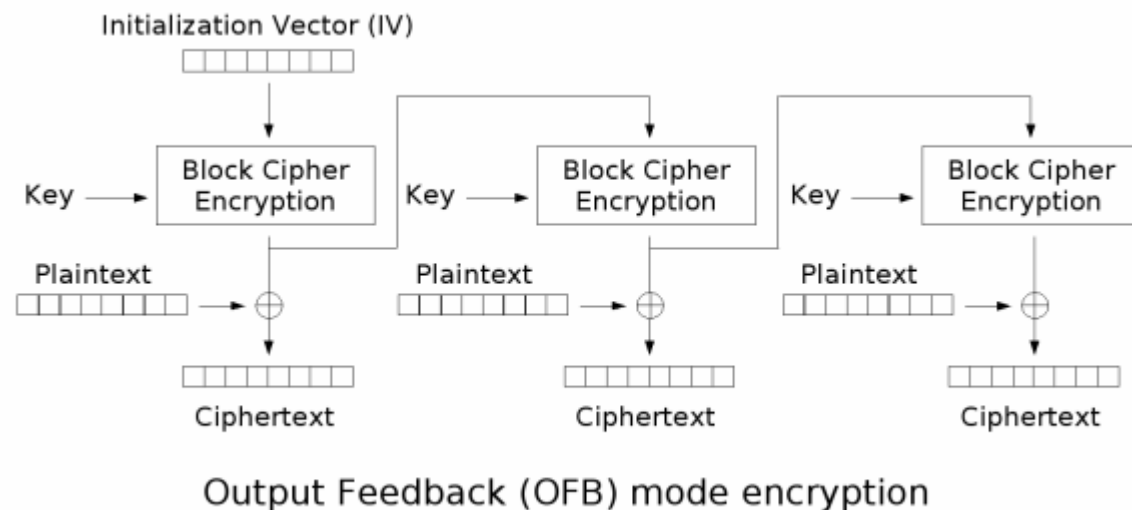
Cipher Block Chaining (CBC) mode encryption

IV oraz Nonce

- Wektor inicjalizujący, wartość początkowa (Initialisation vector)
 - P_0 i C_0 w trybie CBC
 - Może być jawny, musi być nieprzewidywalny
 - Nie może być licznikiem
- Nonce (*number used once*)
 - Wartość unikalna
 - Użyta tylko jeden raz z danym kluczem
 - Może być licznikiem lub numerem wiadomości

Output Feed Back (OFB)

- Ciąg kluczowy generowany niezależnie
 - Szyfrowanie przez XOR
 - $K_0 := IV$
 - $K_i := E(K, K_{i-1})$
 - $C_i := P_i \oplus K_i$



Tryb licznikowy (CTR)

- Ciąg kluczowy generowany niezależnie
 - Szyfrowanie przez XOR
 - $K_i := E(K, \text{Nonce} \parallel \text{licznik})$
 - $C_i := P_i \oplus K_i$
- Możliwe przetwarzanie równoległe
- *Nonce* krytyczne dla bezpieczeństwa

Tryby specjalne

- XTS-AES (SP 800-38E)
 - Tryb do szyfrowania nośników danych (np. Dysków) ze swobodnym dostępem do bloków
 - Pracuje na sektorach, większych niż jeden blok
 - Standard IEEE 1617-2007
- Tryby kombinowane (*encryption-authentication*)
 - CCM (SP800-38C)
 - GCM (SP800-38D)
 - Szybszy od CCM
 - OCB 2.0
 - Najszybszy, opatentowany

Kryptograficzne funkcje skrót

- Funkcja skrót (*hash function*)
 - Dane dowolnej długości na wejściu – m
 - Skrót o stałej długości na wyjściu – $h(m)$
 - Typowe długości skrót - 128-1024 bity
- Unikalna reprezentacja wiadomości
- Zastosowanie
 - Kontrola integralności
 - Podpis cyfrowy
 - Funkcja jednokierunkowa

Wymagania wobec funkcji skrótu

- Odporność na kolizje (*collision resistance*)
 - Kolizja – dowolne m_1, m_2 gdzie $h(m_1)=h(m_2)$
- *First preimage resistance*
 - Obliczenie $x=h(m)$ jest łatwe, znalezienie m na podstawie x jest praktycznie niemożliwe
 - Jednokierunkowość (*one-way function*)
- *Second preimage resistance*
 - Dla wybranego m_1 znalezienie m_2 gdzie $h(m_1)=h(m_2)$ jest praktycznie niemożliwe

Funkcje skrótu w praktyce

	MD5	SHA1	SHA-224	SHA-256	SHA-384	SHA-512	SHA3
Długość	128	160	224	256	384	512	224-512
Kolizje	24	60	112	128	192	256	?
Preimage	116	160	224	256	384	512	?
2nd preimage	123	105-160	201-224	201-256	384	394-512	?

Zasady doboru funkcji skrótu:

- W nowych aplikacjach
 - funkcje z grupy SHA2
 - przygotować na SHA3 (*length extension attack*)
- SHA1 powinno być zastępowane po 2010 roku
- MD5 powinno być zastępowane od 1999 roku

Powyższe dotyczy zastosowań, gdzie konieczna jest długoterminowa odporność na kolizje i inne ataki (np. podpis elektroniczny). W niektórych zastosowaniach (np. SSL) nawet MD5 może zapewniać wystarczający poziom bezpieczeństwa.

Funkcje uwierzytelniające

- MAC (*Message Authentication Code*)
 - Ochrona integralności
 - Ochrona autentyczności pochodzenia
 - Autentyczność zapewnia tajny klucz K
 - Nadawca generuje MAC i dołącza do wiadomości
 - $m, T := \text{MAC}(K, m)$ T – "*authentication tag*"
 - Odbiorca generuje MAC i porównuje z otrzymanym
 - $m, T =? \text{MAC}(K, m)$
 - Funkcja skrótu sparametryzowana tajnym kluczem

Funkcje MAC w praktyce

- HMAC (FIPS 198a)
 - $h(K \oplus a || h(K \oplus b || m))$
 - Najbardziej rozpowszechniony z SHA1
 - Zalecany z SHA-256
- CMAC (NIST SP 800-38A)
 - CBC-MAC, wykorzystuje AES-128, 192, 256
 - Zalecany z AES-256
- Realna złożoność $2^{n/2}$
 - Wymagane min. 128 bitów

Wymiana kluczy kryptograficznych

Skąd się biorą klucze?

- Liczby losowe w kryptografii
 - Krytyczne dla bezpieczeństwa
 - Klucze kryptograficzne, ciągi kluczowe, IV, nonce, identyfikatory...
- Źródła liczb losowych
 - Generatory pseudolosowe (PRNG)
 - Istotna jakość ciągu wyjściowego
 - Tylko PRNG zaprojektowane do celów kryptograficznych
 - Konieczne zasilenie (*seed*)
 - Generatory sprzętowe (*hardware RNG*)
 - Koszt, wydajność

Entropia informacji

- Entropia informacji
 - suma wystąpień znaków w tekście w stosunku do wszystkich możliwych kombinacji
- Teksty rzeczywiste
 - Niska entropia (~ 4)
- Ciągi losowych bajtów
 - Wysoka entropia (~ 8)
- Dobry szyfr produkuje kryptogram o wysokiej entropii
 - Nawet jeśli entropia klucza i tekstu jawnego jest niska

Od hasła do klucza

- Niska złożoność haseł użytkowników
 - Średnio ok. 40 bitów entropii
- Techniki zwiększania złożoności
 - Modulowany skrót hasł do uwierzytelnienia
 - Ustawienie hasła: $salt, p := h(hasło || salt)$
 - Logowanie: $h(hasło || salt) = p?$
 - Iterowany skrót do szyfrowania
 - PKCS#5, Scrypt

Metody wymiany klucza

- Bezpieczne obliczenie wspólnego klucza
 - Algorytm Diffie-Hellman (1976)
 - Krzywe eliptyczne (ECDH)
- Kryptografia z kluczem publicznym
 - Algorytm RSA

Wymiana klucza a uwierzytelnienie

- Atak *man-in-the-middle*
 - Niemożliwe rozwiązanie technikami kryptograficznymi
- Środki organizacyjne
 - Zaufana trzecia strona
 - PKI

Protokół kryptograficzny w praktyce

1) Przedstawienie się (*identification*)

- deklaracja tożsamości

2) Uwierzytelnienie (*authentication*)

- potwierdzenie zadeklarowanej tożsamości
 - podpis elektroniczny (RSA, DSA), lub
 - wspólne hasło (shared secret)
- W obu przypadkach konieczne dodatkowe mechanizmy organizacyjne!

Protokół kryptograficzny w praktyce

3) Wymiana klucza (key exchange, key agreement)

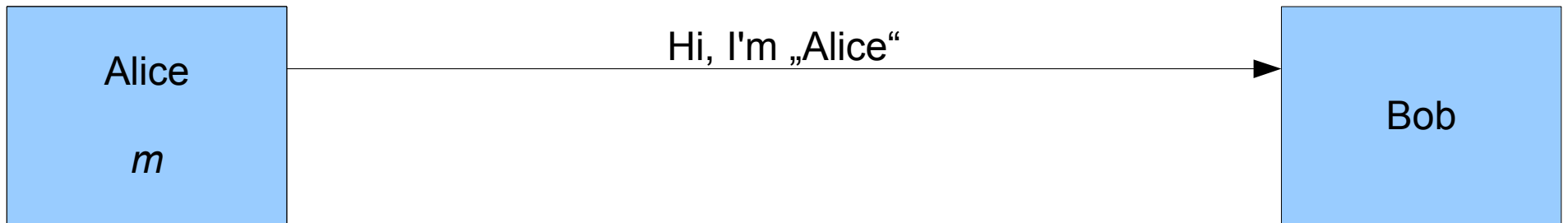
- wymiana losowego klucza sesyjnego (RSA), lub
- ustalenie wspólnego klucza sesyjnego (DH)

4) Transmisja danych

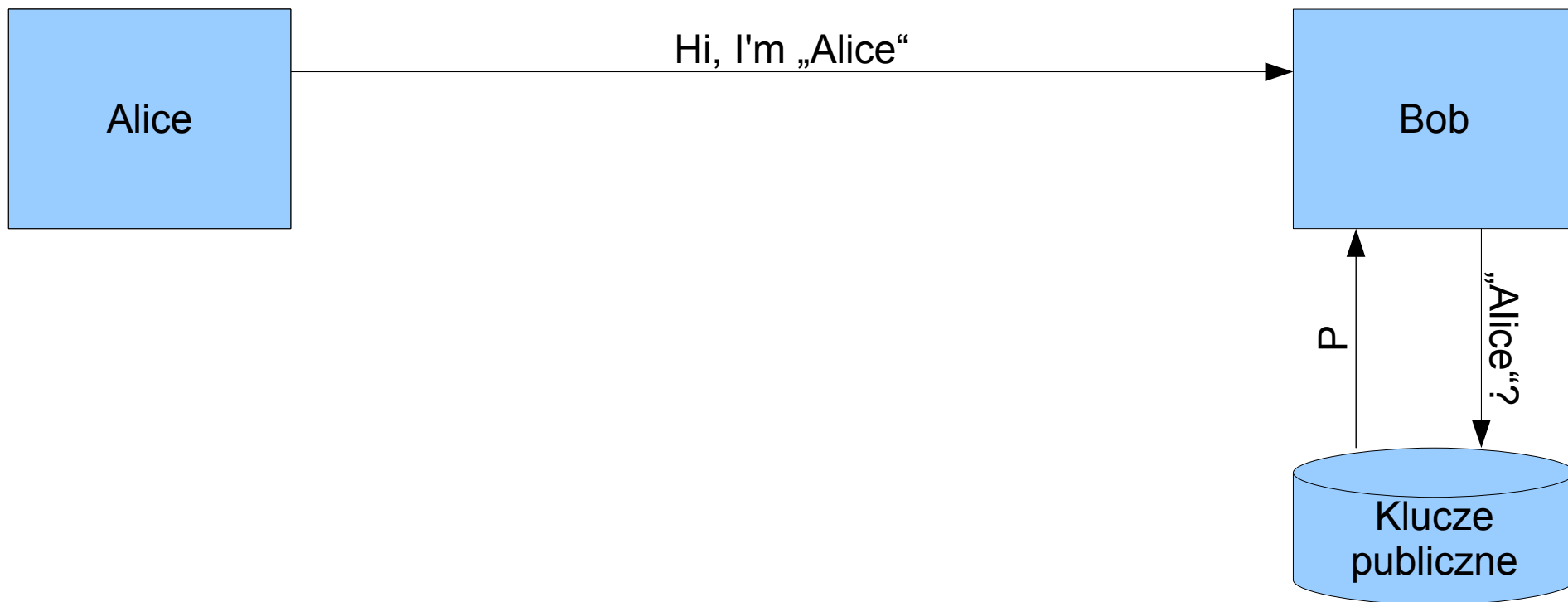
5) Zarządzanie połączeniem

- Ustanawianie nowych kluczy co X godzin lub Y bajtów (*renegotiation*)

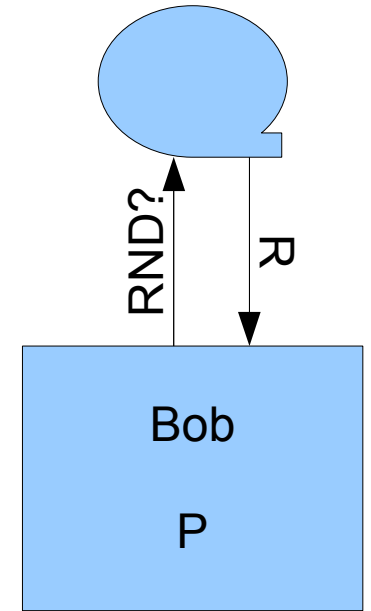
Identyfikacja



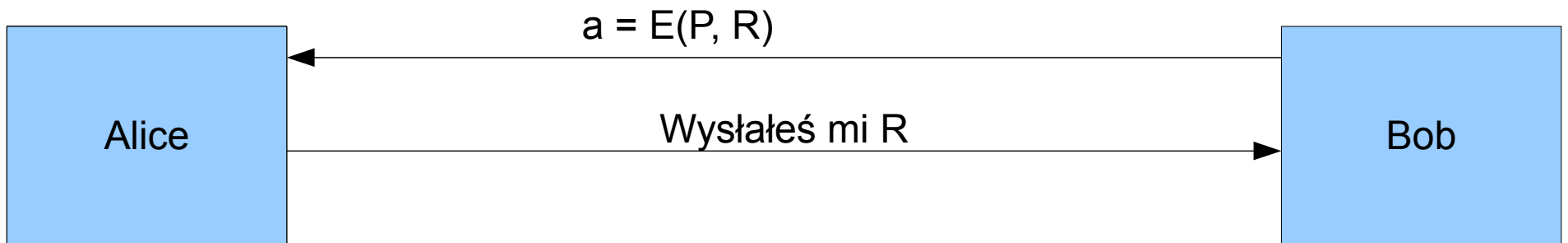
Identyfikacja



Uwierzytelnienie



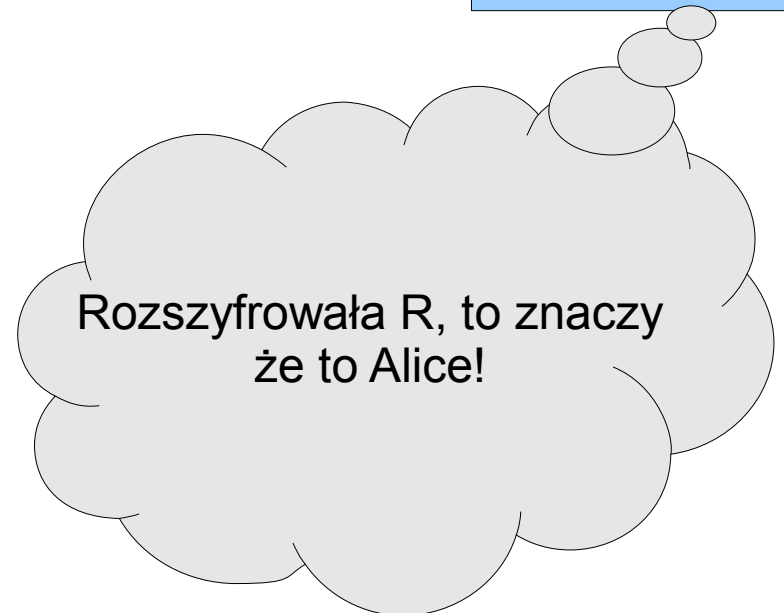
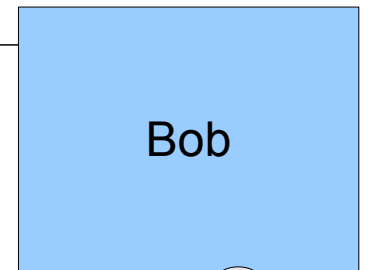
Uwierzytelnienie



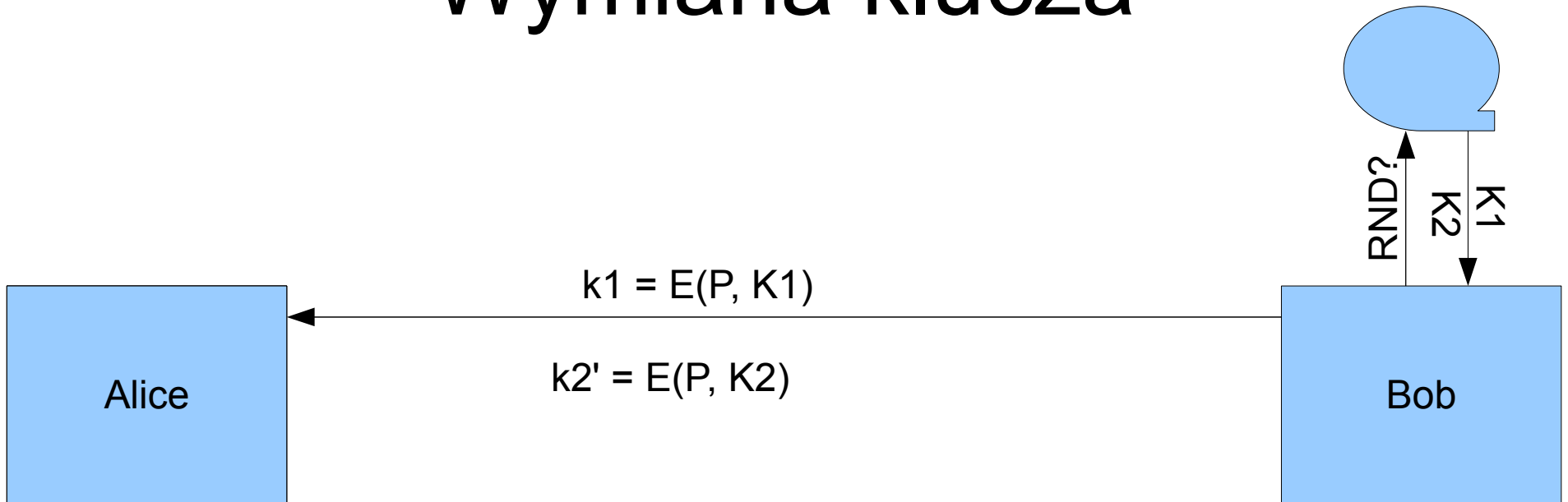
Uwierzytelnienie



Teraz wymieńmy się
kluczami



Wymiana klucza



Transmisja danych

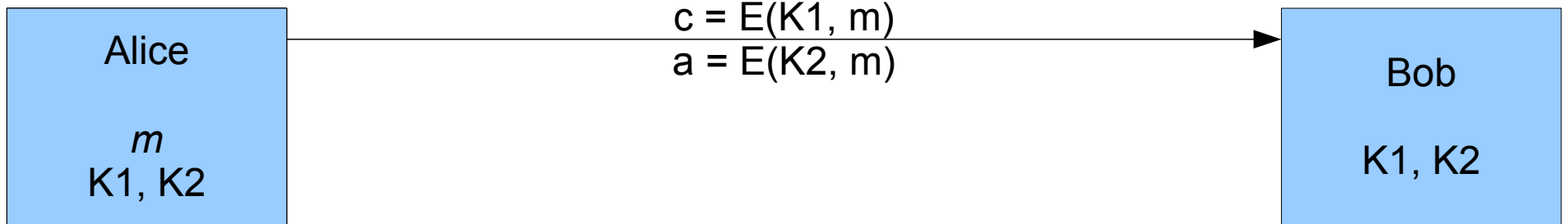
Alice

m
K1, K2

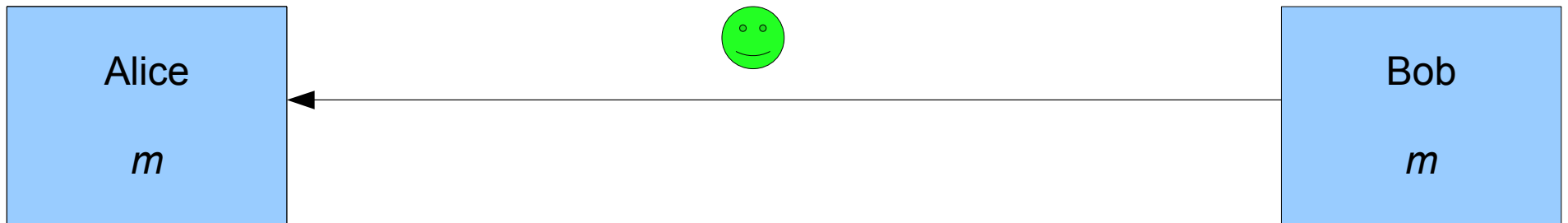
Bob

K1, K2

Transmisja danych



Transmisja danych



Kontakt z autorem:

pawel.krawczyk@hush.com

Tel. +48-602-776959

Prezentacja udostępniona na licencji Creative Commons BY-NC-SA
(„uznanie autorstwa”, „użycie niekomercyjne”, „na tych samych warunkach”)

<http://creativecommons.org/licenses/by-nc-sa/3.0/pl/>

Kontakt z autorem:

pawel.krawczyk@hush.com

Tel. +48-602-776959

Prezentacja udostępniona na licencji Creative Commons BY-NC-SA
(„uznanie autorstwa”, „użycie niekomercyjne”, „na tych samych warunkach”)

<http://creativecommons.org/licenses/by-nc-sa/3.0/pl/>

Kryptografia i mechanizmy bezpieczeństwa

Paweł Krawczyk

Kontakt z autorem:

pawel.krawczyk@hush.com

Tel. +48-602-776959

Prezentacja udostępniona na licencji Creative Commons BY-NC-SA
(„uznanie autorstwa”, „użycie niekomercyjne”, „na tych samych warunkach”)

<http://creativecommons.org/licenses/by-nc-sa/3.0/pl/>

2

Literatura:

<http://ipsec.pl/>

<http://securitystandard.pl/>

<http://csrc.nist.gov/>

Niels Ferguson, Bruce Schneier, t Tadayoshi Kohno
"Cryptography engineering"

Konspekt

- Funkcje kryptografii
- Podstawowe operacje kryptograficzne
- Ataki kryptoanalityczne
- Podstawowe techniki kryptograficzne
- Zarządzanie kluczami

Funkcje kryptografii

- Poufność
 - Szyfrowanie
- Integralność
 - Funkcje skrótu
- Rozliczalność
 - Protokoły uwierzytelniania, integralność logów
- Anonimowość
 - Dowód z wiedzą zerową, szyfrowanie
- Dostępność
 - Kontrola dostępu

Terminologia

- Tekst jawny, otwarty (*plain, clear text* - P)
- Szyfrogram, kryptogram (*cipher text* – C)
- Szyfrowanie – deszyfrowanie (rozszyfrowanie)
- Klucz (*key*)
 - Tajny parametr – realizuje kontrolę dostępu do tekstu jawnego
- Atak kryptoanalityczny
 - Teoretyczny
 - Praktyczny

5

Słowa "kod" i "szyfr" są często stosowane jako synonimy. Szyfr różni się od kodu tym, że posiada dodatkowy, tajny parametr czyli klucz szyfrujący.

Terminologia

- Algorytm kryptograficzny
 - Funkcja bezpieczeństwa
 - DES, AES, MD5, SHA-1, RSA...
- Protokół kryptograficzny
 - Operacja
 - ESP (IPSec), SSL Record Protocol (TLS)
- System kryptograficzny (*cryptosystem*)
 - Funkcja biznesowa
 - PGP, X.509

6

Algorytmy są podstawowymi jednostkami służącymi do budowy protokołów kryptograficznych. Algorytm realizuje określoną funkcję – np. poufności.

Protokół wykorzystuje kilka algorytmów w celu zapewnienia szeregu pożądaných funkcji bezpieczeństwa, służy jednak realizacji jednej określonej operacji – np. szyfrowana transmisja danych.

System kryptograficzny korzysta z wielu protokołów i algorytmów zapewniając kompletny i spójny zestaw operacji niezbędnych do realizacji określonych potrzeb biznesowych – np. bezpieczna wymiana poczty elektronicznej musi obejmować transmisję danych, wymianę kluczy, uwierzytelnienie stron itd.

Zasada Kerckhoffsza

- Zasada Kerckhoffsza
 - Auguste Kerckhoffs 1883
 - Bezpieczeństwo systemu powinno polegać na tajności klucza, a nie systemu
 - Claude Shannon
 - „Wróg zna szczegóły systemu”

7

Oryginalne brzmienie praw Kerckhoffsza

- system musi być praktycznie, a lepiej matematycznie, do złamania
- nie można żądać by system był utajniony, w razie wpadnięcia w ręce nie przyjaciela powinien pozostać bezpiecznym
- musi nadawać się do korespondencji telegraficznej
- musi być przenośny, a jego używanie nie może wymagać zaangażowania wielu osób
- system powinien być łatwy w użyciu, nie powinien wymagać dużego wysiłku umysłowego ani pamiętania długich zbiorów reguł

Jawność algorytmów kryptograficznych

- Algorytmy jawne
 - Niezależna weryfikacja poprawności
 - Cryptology ePrint Archive (IACR), ArXiv.org
 - Większość algorytmów cywilnych jest jawna
- Algorytmy niejawne
 - Dodatkowe zabezpieczenie, utrudnia analizę
 - Tylko jeśli skuteczne ograniczenie dostępu do implementacji jest możliwe
 - Nie dotyczy większości implementacji komercyjnych

8

Niemal wszystkie stosowane obecnie algorytmy cywilne są jawne. Warunkiem udziału w konkursach mających wyłonić standardy kryptograficzne (AES, SHA-3, NESSIE) jest jawność.

Niejawne algorytmy są stosowane w zastosowaniach rządowych i wojskowych (np. polski NASZ).

Niejawność sprawdza się tylko wtedy gdy dostęp do implementacji jest ograniczony, zaś ona sama jest zabezpieczona przed penetracją.

Większość komercyjnych, niejawnych algorytmów została odtworzona i niekiedy złamana gdy tylko udało się uzyskać dostęp do implementacji (RC4, DVD CSS, MS-DRMv2, GSM – A5, COMP128), Mifare CRYPTO-1.

Niejawność algorytmu nie może stanowić podstawy jego bezpieczeństwa.

Standardy kryptograficzne

- Odpowiedzialność projektanta systemów
- Wybór technik kryptograficznych
 - 100% oparcia w standardach i normach
 - Regulacje polskie i europejskie (uodo, uope, uooin, bankowe)
 - NIST Computer Security Resources Center
 - NIST Federal Information Processing Standards (FIPS)
 - NIST Special Publications (zalecenia)

9

NIST SP 800-53 "Recommended Security Controls for Federal Information Systems and Organizations"

NIST SP 800-57 "Recommendation for Key Management"

IETF RFC (Request For Comments)

RSA PKCS (Public Key Cryptography Standards)

Standardy kryptograficzne

- Wybór implementacji kryptograficznych
 - Wbudowane w system lub framework aplikacji
 - Microsoft Cryptographic API (CAPI)
 - Biblioteki kryptograficzne
 - Komercyjne, otwarte
 - Certyfikaty bezpieczeństwa

10

Certyfikacja bibliotek i modułów sprzętowych:

NIST FIPS 140-2

Cryptographic Algorithm Validation Program (CAVP)

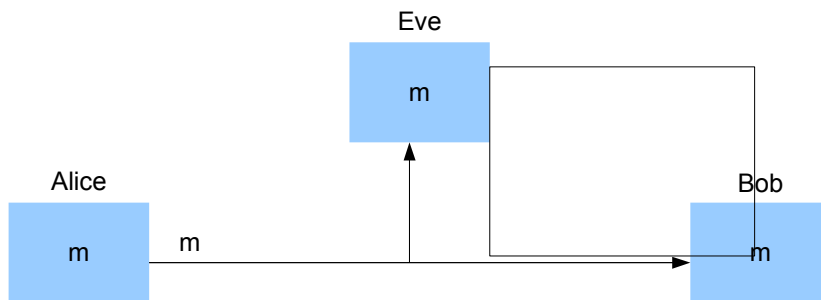
Cryptographic Module Validation Program (CMVP)

Common Criteria, ITSEC

CWA 14169

Podstawowe operacje kryptograficzne

Atak: podsłuch



12

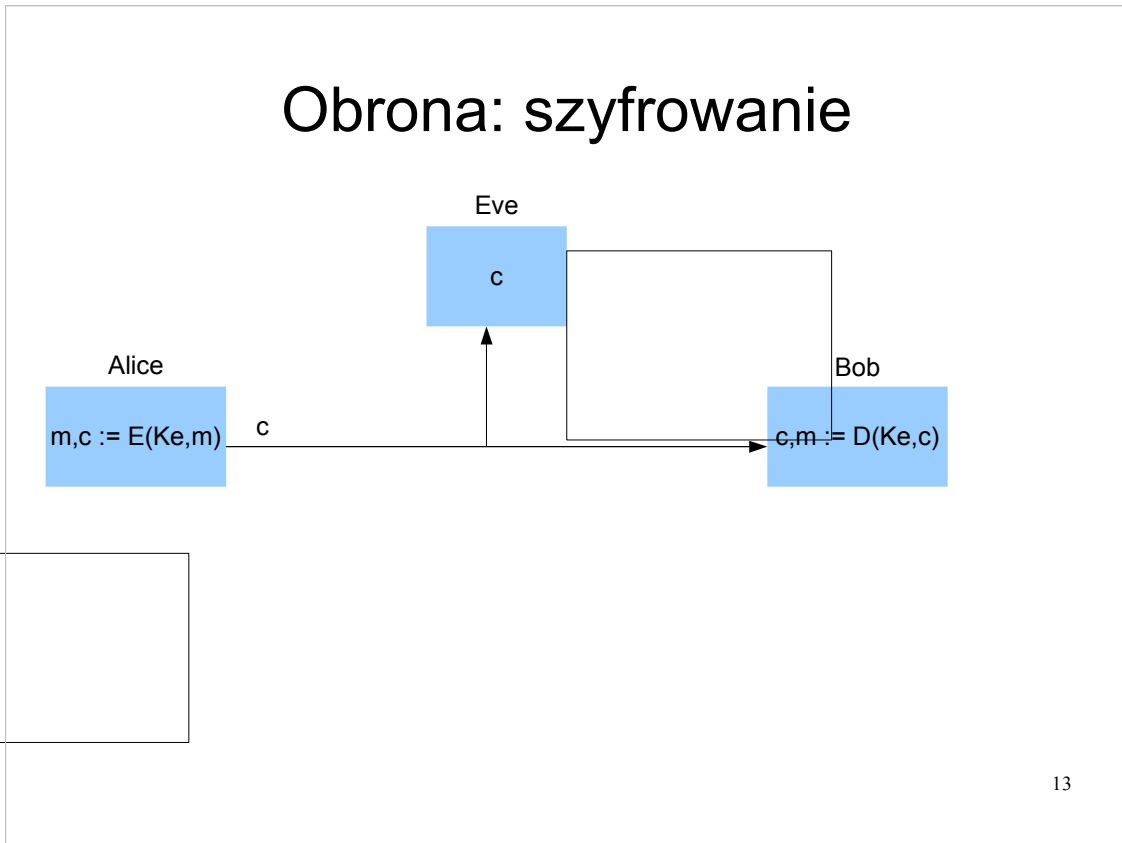
Alice i Bob to tradycyjnie przyjęte w literaturze kryptograficznej imiona stron komunikacji. Eve ("evil") występuje zawsze w roli atakującego – chce podsłuchać, sfałszować lub w inny sposób zaszkodzić oryginalnej komunikacji między Alice i Bobem.

Schemat pokazuje transmisję wiadomości (m jak "*message*") w sposób jawny, bez użycia mechanizmów kryptograficznych.

W tym scenariuszu Eve może bez ograniczeń podsłuchiwać lub podrabiać wiadomości między Alice i Bobem.

Eve wykonuje atak pasywny przeciwko poufności wiadomości.

Obrona: szyfrowanie



13

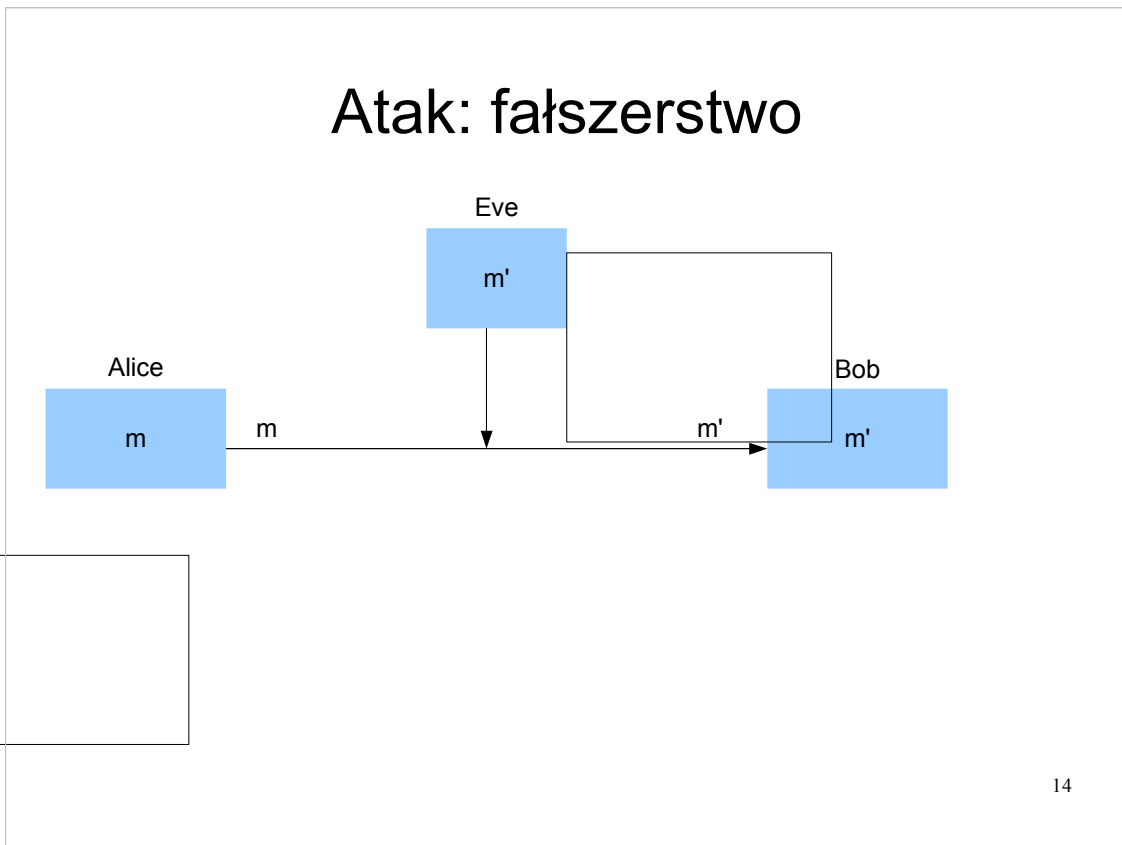
Alice szyfruje (E) wiadomość m przy pomocy klucza (K_e). Wynikiem jest kryptogram c , który jest następnie wysyłany do Boba.

Bob – dysponujący kluczem K_e – może rozszyfrować (D) kryptogram i otrzyma oryginalną wiadomość m .

Eve zgodnie z zasadą Kerckhoffs'a może znać szczegóły algorytmu, ale dopóki nie zna klucza K_e nie powinna mieć praktycznej możliwości odtworzenia wiadomości m .

Proszę zwrócić uwagę, że pomijamy zupełnie kwestię jak Alice i Bob mogą mieć ten sam tajny klucz K_e , skoro generalnie mają problem z bezpieczną łącznością?

Atak: fałszerstwo

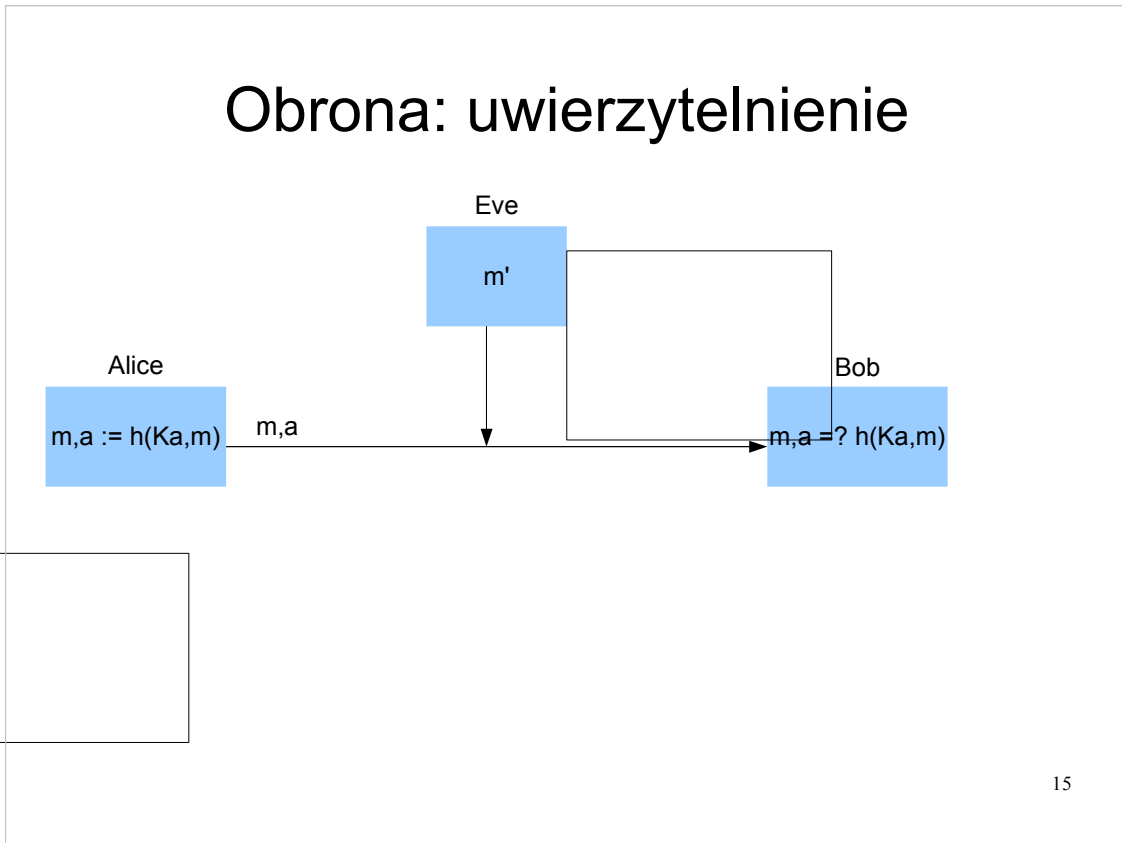


14

Znowu scenariusz z łącznością jawną. Eve tym razem uniemożliwia doręczenie oryginalnej wiadomości m i zamiast niej wysyła do Boba wiadomość podrobioną (m').

Tym razem jest to atak aktywny przeciwko integralności wiadomości.

Obrona: uwierzytelnienie



Alice wprowadza kod uwierzytelniający wiadomości (MAC – Message Authentication Code).

Autentyczność pochodzenia jest zapewniana przez kolejny klucz tajny (Ka), inny niż do szyfrowania, które tutaj pominięto.

MAC, dla tej samej wiadomości i tego samego klucza Ka , da zawsze ten sam wynik (a). Kod a może być przesyłany w postaci jawnej – jego znajomość nie umożliwia odtworzenia wiadomości m .

Technicznie MAC jest realizowany przy pomocy funkcji skrótu (h).

Eve nadal może wysłać fałszywą wiadomość m' .

Jednak Bob powtarza operację MAC dla otrzymanej wiadomości i porównuje jej wynik z otrzymanym kodem a .

Eve mogłaby także wysłać wcześniej przechwycone wiadomości wraz z ich kodami MAC (replay attack). Alice i Bob chronią się przed tym numerując wiadomości (sequence number).

Koncepcja klucza publicznego

- Kryptografia symetryczna
 - Alice i Bob stosują ten sam klucz (K_e , K_a)
 - Źle skalowalny (10 os. - 45 kluczy, 20 – 190 itd)
- Kryptografia asymetryczna
 - Alice ma teraz
 - Klucz do szyfrowania (publiczny – jawny)
 - P_{Alice} (*public*)
 - Klucz do deszyfrowania (prywatny – tajny)
 - S_{Alice} (*secret*)
- Public Key (PK) Cryptography (PKC)

Szyfrowanie kluczem publicznym



17

Alice szyfruje (E) wiadomość m kluczem publicznym Boba (P_{Bob}). W kryptografii z kluczem publicznym do szyfrowania używamy klucza publicznego adresata.

Bob deszyfruje otrzymany kryptogram c przy pomocy swojego klucza prywatnego (S_{Bob}).

Jeszcze o kluczach publicznych

- Wiadomość zaszyfrowana P_x
 - Tylko posiadacz S_x może ją odczytać
 - Nawet nadawca jej nie odczyta
 - Jeśli np. zgubi oryginał
 - W praktyce – nadawca używa i P_x i swojego P
- Znacząco zredukowana liczba kluczy
 - Każdy publikuje tylko swój klucz

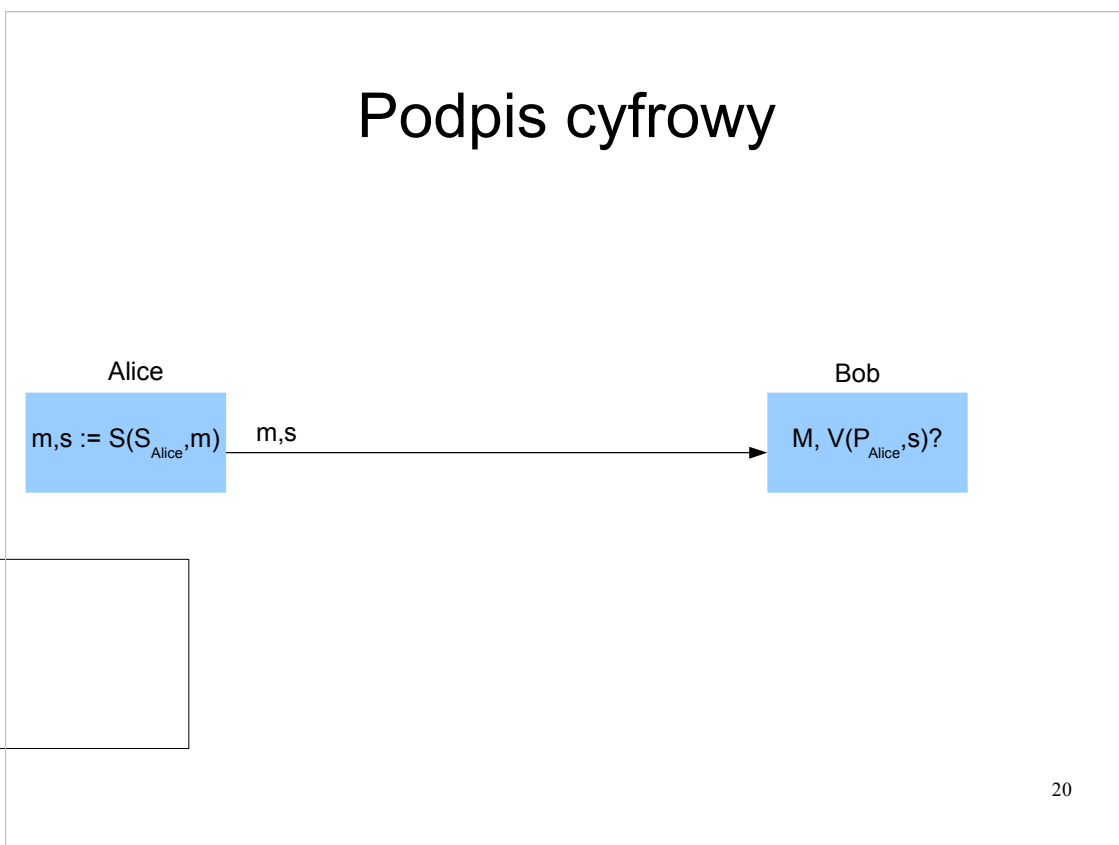
18

Kluczowa obserwacja – tylko posiadacz klucza prywatnego pasującego do użytego klucza publicznego może rozszyfrować wiadomość. Nawet Alice nie będzie w stanie odszyfrować kryptogramu, jeśli zgubi oryginalną wiadomość!

Szyfrowanie hybrydowe

- Wady PKC
 - Wymagające obliczeniowo i pamięciowo
 - O rząd wielkości wolniejsze niż szyfry symetryczne
- Rozwiązanie
 - Wiadomość m szyfrujemy K_e
 - Duża porcja danych, szybki szyfr
 - K_e szyfrujemy P_x
 - Mała porcja danych, wolny szyfr
 - Odbiorca deszyfruje K_e swoim S_x
 - Używając K_e deszyfruje m

Podpis cyfrowy



Operacja podpisu cyfrowego również korzysta z pary kluczy S, P ale należących do osoby składającej podpis.

Podpis jest składany przy pomocy klucza prywatnego Alice. Daje to wskazówkę, że tylko Alice mogła stworzyć tę wiadomość (niezaprzeczalność).

Bob weryfikuje podpis przy pomocy klucza publicznego (P) nadawcy czyli Alice. Ponieważ jej klucz jest jawny, każdy może łatwo zweryfikować autentyczność podpisanego dokumentu.

Technicznie podpis realizuje się przez zaszyfrowanie kluczem prywatnym (S) skrótu wiadomości.

Weryfikacja polega na próbie odszyfrowania przy pomocy klucza publicznego (P). Powodzenie operacji dowodzi, że podpis złożono pasującym kluczem S .

Terminologia e-podpisowa

- Pojęcia techniczne i prawne – kolizje
- Podpis cyfrowy
 - Pojęcie inżynierskie
 - Poświadczenie autentyczności i integralności danych
 - Różne zastosowania
- Podpis elektroniczny
 - Pojęcie prawno-inżynierskie
 - Związany z osobą fizyczną

21

ISO 7498-2:1989 – „Digital signature”

- Digital signature: data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source of the data unit and protect against forgery, e.g. by the recipient
- Podpis pod danymi, dokumentem – lub operacja „podpisu” przy logowaniu do systemu i inne tego typu – Te zastosowania (z wyłączeniem „podpisu pod dokumentem”) obejmuje rozszerzenie `keyUsage=digitalSignature`

Dyrektywa UE 1999/93/EC – „Electronic signature”

- Electronic signature means data in electronic form which are attached to or logically associated with other electronic data and which serve as a method of authentication

Tylko podpis pod konkretną treścią

- To zastosowanie obejmuje rozszerzenie `keyUsage=nonRepudiation`

Autentyczność kluczy

- Klucz Ka lub Px zapewnia autentyczność danych
 - Ale co zapewnia autentyczność Ka i Px?
 - Konieczne rozwiązania organizacyjne
- Poświadczenie kluczy
 - Wzajemne (mesh)
 - Zaufana trzecia strona (TTP – *Trusted Third Party*)
 - Urzędy certyfikacji, centra certyfikacji (CA – *Certifying authority*)
 - Informacja o aktualności kluczy

PKI

- Infrastruktura klucza publicznego
 - PKI – *Public Key Infrastructure*
 - Środki techniczne, organizacyjne i prawne
 - Systemy kryptograficzne
 - TTP świadczące usługi certyfikacyjne
 - Osadzenie w prawie

Ataki kryptoanalityczne

Cel ataku na szyfr

- Cel szyfru
 - Utrzymanie złożoności ataku na poziomie przeszukiwania wszystkich kluczy (*brute force, exhaustive*)
- Pośredni cel ataku
 - Obniżenie złożoności poniżej tej wartości
- Bezpośredni cel ataku
 - Uzyskanie dostępu do wiadomości

Obiekt ataku

- Ataki na algorytmy
 - Słabości w algorytmach – Enigma, Lorenz, PURPLE, RC4
- Ataki na implementacje
 - WEP (RC4), PPTPv1 (RC4), OpenSSL (RSA)
- Ataki na użytkownika
 - Kleptografia, "*rubber-hose cryptanalysis*"

Wiedza atakującego

- Tylko kryptogram (*ciphertext only*)
 - Zawsze
- Znany tekst jawny (*known-plaintext*)
 - Przewidywalny tekst jawny, tekst podeślany
- Wybrany tekst jawny (*chosen plaintext*)
- Wybrany tekst jawny i zaszyfrowany (*chosen ciphertext*)

Przykłady ataków

- Paradoks dnia urodzin (*birthday paradox*)
 - W grupie 23 osoby prawdopodobieństwo urodzin tego samego dnia przekracza 50% ("kolizja")
 - Wśród identyfikatorów transakcji z przedziału 2^{64} – kolizje po 2^{32} transakcjach
 - W zbiorze N elementów – 50% kolizji po \sqrt{N}
 - Duże znaczenie dla
 - Identyfikatorów wiadomości (*anti-replay*)
Wartości, które nie mogą się powtarzać (niektóre klucze)

Przykłady ataków

- Spotkanie w środku (*meet-in-the-middle*, *collision attack*)
 - Pregenerowanie "słownika"
 - 2^{32} par: tekst jawny \rightarrow kryptogram
 - Śledzenie komunikacji
 - Kolizja prawdopodobna już po 2^{32} kryptogramach
 - Złożoność po ataku: $2^{32} + 2^{32} = 8,6 \times 10^9$
 - Oryginalna złożoność: $2^{64} = 1,8 \times 10^{19}$

Realna siła algorytmów

- Idealnie – równa długości klucza
 - Klucz 2^{128} bitów $\rightarrow 2^{128}$ kroków do złamania klucza
- Znane ataki mogą zmniejszać złożoność
 - 2TDES – klucz 112 bitów – siła ok. 80 bitów
 - 3TDES – klucz 168 bitów – siła ok. 112 bitów
 - DES-X – klucz 184 bity – siła ok. 118 bitów

Podstawowe techniki kryptograficzne

Szyfr blokowy

- Pracuje na porcjach danych o stałej długości
 - Bloki - obecnie min. 128 bitów (16 bajtów)
- Symetryczny
 - Ten sam klucz do szyfrowania i deszyfrowania
 - Obecnie min. 128 bitów (16 bajtów)
- Stałe przekształcenie
 - Ten sam algorytm dla tego samego K_e i tego samego P da ten sam C
- Standardowy interfejs
 - Ustaw klucz, szyfruj, deszyfruj

Wymagania wobec szyfrów

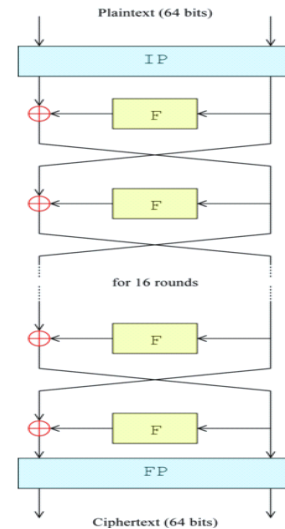
- Efekt lawinowy (*avalanche effect*)
 - Zmiana jednego bitu tekstu jawnego lub klucza zmienia wszystkie bity kryptogramu
- Rozproszenie (*diffusion*)
 - Zatarcie charakterystyki tekstu jawnego w kryptogramie
- Przemieszanie (*confusion*)
 - Zatarcie związku między kluczem a kryptogramem

Szyfry blokowe w praktyce

- DES (Data Encryption Standard)
 - Blok 64 bity, klucz 56 bitów
 - Nie używać
- 3DES
 - Blok 64 bity, klucz 168 bitów
- AES (Advanced Encryption Standard)
 - Blok 128 bitów, klucz 128, 192, 256 bitów
 - Aktualnie zalecany standard cywilny
- Serpent, Twofish, MARS, RC6

DES

- Standard od 1975
- Niezalecany od 1999 (NIST)
- Brak ataków praktycznych lepszych niż *brute force*
- Wady
 - Krótki klucz (RSA DESCHALL)
 - Krótki blok (kolizje)
 - Niska wydajność



35

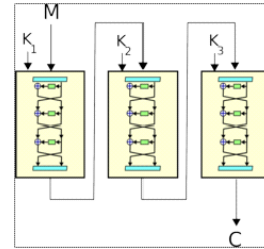
DES w dokumentach oficjalnych jest także znany pod nazwą DEA (Data Encryption Algorithm).

DES nadal jest stosowany w niektórych systemach bankowych. Wymienione wady mogą nie mieć znaczenia praktycznego jeśli dane wejściowe mają niewielką długość (np. numer karty kredytowej), krótki czas życia i nie jest wymagana wysoka wydajność.

DES jest algorytmem projektowanym pod kątem implementacji sprzętowej, stąd niska wydajność w oprogramowaniu.

3DES

- Szyfrowanie-deszyfrowanie-szyfrowanie
 - 3DES-EDE
 - Długość klucza $3 \times 56 = 168$ bitów
- Dopuszczalny pod warunkiem używania trzech niezależnych kluczy
- Wady
 - Krótki blok (kolizje)
 - Niska wydajność



36

3DES jest prawdopodobnie najbardziej rozpowszechnionym szyfrem. Cieszy się zaufaniem (35 lat odporności na kryptoanalizę).

Powszechnie stosowany w bankowości, zastosowaniach rządowych i wojskowych.

Polskie szyfratory do informacji niejawnej CompCrypt stosują 3DES do poziomu "tajne". Na poziom "ściśle tajne" jest dopuszczony niejawny algorytm NASZ-1, stanowiący zmodyfikowaną wersję 3DES.

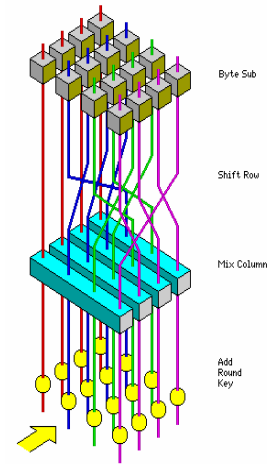
W praktyce dużą wadą 3DES jest niska wydajność – 3x niższa od DES, który i tak jest dość wolny w implementacji programowej.

NIST SP 800-57 dopuszcza 3DES (TDEA) pod warunkiem używania trzech różnych kluczy.

NIST SP 800-67 Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher

AES

- Advanced Encryption Standard (AES)
 - Rijndael
- Bardzo wydajny
- Wg niektórych niski margines bezpieczeństwa



37

AES jest ustandaryzowaną wersją szyfru Rijndael.

Rijndael dopuszczał klucze o dowolnej długości w określonym przedziale. AES, dla zachowania interoperacyjności, dopuszcza tylko klucze 128, 192 i 256 bitów.

AES jest szyfrem bardzo wydajnym w implementacjach programowych.

NIST SP 800-57 dopuszcza go dla wszystkich długości kluczy.

Inne szyfry

- Serpent
 - Wysoki margines bezpieczeństwa, wolniejszy od AES, darmowy
- Twofish
 - Wydajny, darmowy
- MARS
 - Darmowy
- RC6
 - Opatentowany (RSA)

Tryby szyfrowania

- Blok 128 bitów = 16 bajtów
 - Podział tekstu jawnego na bloki
 - Uzupełnianie bloków (*padding*)
- Różne sposoby podziału na bloki
 - Tryby szyfrowania (*block cipher modes*)
 - Różne istotne konsekwencje dla bezpieczeństwa
 - Dodatkowe parametry
- Standardowy interfejs programistyczny
 - `DES_cbc_encrypt()`, `DES_cbc_decrypt()`

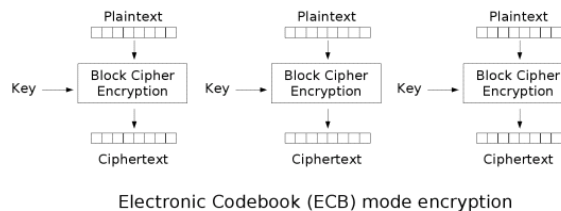
39

Obszerna lista standardowych trybów szyfrowania wraz z uwagami dotyczącymi ich stosowania w praktyce została opublikowana przez NIST w SP 800-38A.

Szereg dodatkowych trybów do zastosowań specjalnych została opublikowana w innych dokumentach z tej serii (SP 800-38B do 800-38E).

Electronic Code Book (ECB)

- Najbardziej intuicyjny
 - P dzielony na bloki po 16 bajtów
 - Każdy blok szyfrowany samodzielnie
 - $C_i := E(K, P_i)$, $P_i := D(K, C_i)$
- Taki sam blok P da taki sam blok C
 - Szereg problemów związanych z bezpieczeństwem



40

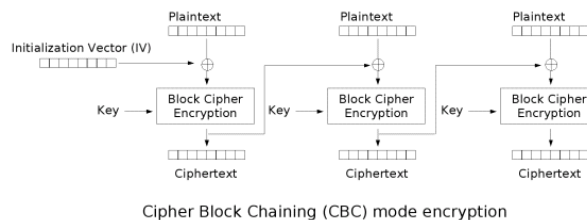
Kryptogram ECB może ujawniać wewnętrzną strukturę danych – jeśli jest to np. zaszyfrowany obrazek (mapa bitowa), składający się z długich sekwencji takich samych bajtów.

Może również ujawniać powtarzające się bloki, zawierające określone wiadomości – np. kwoty przelewu lub komendy systemowe. Śledząc transmisję atakujący może zbudować specyficzną "książkę kodową" i bez znajomości klucza analizować jakie dane są przesyłane w danym momencie.

Zaletą ECB jest możliwość swobodnego dostępu (zmian) w kryptogramie zapisanym np. na dysku.

Cipher Block Chaining (CBC)

- Każdy blok wejściowy jest dodawany do poprzedniego kryptogramu
 - Operacja XOR
 - $C_i := E(K, C_{i-1} \oplus P_i)$
 - $P_i := D(K, C_i) \oplus C_{i-1}$
- Najbardziej rozpowszechniony tryb



41

Tryb CBC jest najbardziej rozpowszechniony w praktyce i może być stosowany z każdym szyfrem blokowym.

Operacja szyfrowania CBC nie może być prowadzona równolegle, co jest wadą.

Modyfikacja jednego bloku tekstu jawnego wpływa na wszystkie kolejne bloki kryptogramu.

Modyfikacja jednego bloku kryptogramu wpływa na dwa bloki tekstu wynikowego.

IV oraz Nonce

- Wektor inicjalizujący, wartość początkowa (Initialisation vector)
 - P_0 i C_0 w trybie CBC
 - Może być jawny, musi być nieprzewidywalny
 - Nie może być licznikiem
- Nonce (*number used once*)
 - Wartość unikalna
 - Użyta tylko jeden raz z danym kluczem
 - Może być licznikiem lub numerem wiadomości

42

Najpopularniejszą metodą generowania IV jest użycie do tego celu liczby losowej. Może być ona jawna i przesłana jako pierwszy blok kryptogramu. W takim przypadku kryptogram wydłuży się o jeden blok.

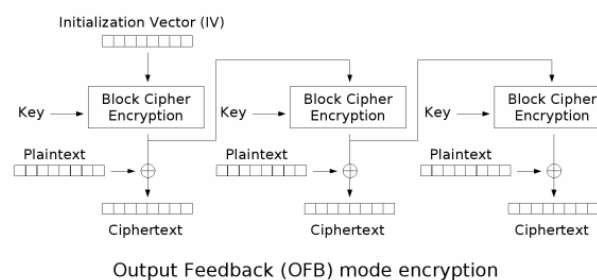
W przypadku bardzo krótkich wiadomości narzut ten może być niepożądany, dlatego IV można wygenerować na podstawie nonce. Może to być liczba o wiele krótsza niż pełny blok szyfru.

Ponieważ niepożądane jest by sam IV był prostym licznikiem, dlatego licznik-nonce można zaszyfrować jako pojedynczy blok uzyskując kryptogram, którego używamy jako IV.

W takim przypadku przesyłanie IV nie jest w ogóle konieczne – wystarczy, że druga strona będzie liczyć odebrane wiadomości.

Output Feed Back (OFB)

- Ciąg kluczowy generowany niezależnie
 - Szyfrowanie przez XOR
 - $K_0 := IV$
 - $K_i := E(K, K_{i-1})$
 - $C_i := P_i \oplus K_i$



43

OFB działa jak szyfr strumieniowy, gdzie generatorem ciągu kluczowego jest szyfr blokowy (dość częsta konstrukcja).

Zaletą jest duża niezależność w generowaniu ciągu kluczowego bez oczekiwania na tekst jawny. Drugą zaletą jest możliwość szyfrowania niepełnych bloków, bo faktyczną operację szyfrowania zapewnia funkcja XOR, która może operować na pojedynczych bajtach czy nawet bitach.

Ze względu na własności funkcji XOR krytyczne dla bezpieczeństwa szyfru jest używanie danego IV tylko jeden raz dla danego klucza szyfrującego.

Tryb licznikowy (CTR)

- Ciąg kluczowy generowany niezależnie
 - Szyfrowanie przez XOR
 - $K_i := E(K, \text{Nonce} || \text{licznik})$
 - $C_i := P_i \oplus K_i$
- Możliwe przetwarzanie równoległe
- *Nonce* krytyczne dla bezpieczeństwa

44

Tryb CTR jest ustandaryzowany w NIST SP 800-38A. Przetwarzanie równoległe umożliwia osiągnięcie bardzo wysokiej wydajności. Jak w przypadku OFB, unikalność *nonce* dla danego klucza K jest krytyczna dla bezpieczeństwa szyfru.

Tryby specjalne

- XTS-AES (SP 800-38E)
 - Tryb do szyfrowania nośników danych (np. Dysków) ze swobodnym dostępem do bloków
 - Pracuje na sektorach, większych niż jeden blok
 - Standard IEEE 1617-2007
- Tryby kombinowane (*encryption-authentication*)
 - CCM (SP800-38C)
 - GCM (SP800-38D)
 - Szybszy od CCM
 - OCB 2.0
 - Najszybszy, opatentowany

Kryptograficzne funkcje skrótu

- Funkcja skrótu (*hash function*)
 - Dane dowolnej długości na wejściu – m
 - Skrót o stałej długości na wyjściu – $h(m)$
 - Typowe długości skrótu - 128-1024 bity
- Unikalna reprezentacja wiadomości
- Zastosowanie
 - Kontrola integralności
 - Podpis cyfrowy
 - Funkcja jednokierunkowa

46

Spotykane są również nazwy polskie: funkcja mieszająca, funkcja haszująca oraz angielskie: *message digest*, *fingerprint* lub po prostu *hash*.

Wymagania wobec funkcji skrótu

- Odporność na kolizje (*collision resistance*)
 - Kolizja – dowolne $m1, m2$ gdzie $h(m1)=h(m2)$
- *First preimage resistance*
 - Obliczenie $x=h(m)$ jest łatwe, znalezienie m na podstawie x jest praktycznie niemożliwe
 - Jednokierunkowość (*one-way function*)
- *Second preimage resistance*
 - Dla wybranego $m1$ znalezienie $m2$ gdzie $h(m1)=h(m2)$ jest praktycznie niemożliwe

47

"Praktycznie niemożliwe" w każdym wypadku oznacza konieczność wyczerpującego przeszukiwania.

Dla funkcji dającej skrót o długości 128 bitów (MD5) złożoność każdego z ataków powinna wynosić 2^{128} .

Atak "dnia urodzin" umożliwia generowanie kolizji przy złożoności $2^{n/2}$, więc dla $n=128$ będzie to 2^{64} .

W 2010 roku NIST określił dolną granicę złożoności na 2^{80} , jak więc widać MD5 jej nie spełnia w zastosowaniach wymagających odporności na kolizje.

RFC 4270, "Attacks on Cryptographic Hashes in Internet Protocols"

NIST SP 800-107

Funkcje skrótu w praktyce

	MD5	SHA1	SHA-224	SHA-256	SHA-384	SHA-512	SHA3
Długość	128	160	224	256	384	512	224-512
Kolizje	24	60	112	128	192	256?	
Preimage	116	160	224	256	384	512?	
2nd preimage	123	105-160	201-224	201-256	384	394-512	?

Zasady doboru funkcji skrótu:

- W nowych aplikacjach
 - funkcje z grupy SHA2
 - przygotować na SHA3 (*length extension attack*)
- SHA1 powinno być zastępowane po 2010 roku
- MD5 powinno być zastępowane od 1999 roku

Powyższe dotyczy zastosowań, gdzie konieczna jest długoterminowa odporność na kolizje i inne ataki (np. podpis elektroniczny). W niektórych zastosowaniach (np. SSL) nawet MD5 może zapewniać wystarczający poziom bezpieczeństwa.

48

Zgodnie z zaleceniami NIST opisanymi we wspomnianej publikacji SP 800-107 oraz - szerzej - w 800-57 funkcja MD5 nie powinna być używana w zastosowaniach związanych z podpisem cyfrowym już od 1999 roku.

W stosunku do SHA-1 taką rekomendację wyznaczono na rok 2010. A ściślej, wyznaczono ją dla wszystkich algorytmów posługując się kryterium zapewnianej odporności mierzonej w bitach. I tak:

- do 2010 roku powinny być używane algorytmy zapewniające minimum 80 bitów - nie powinny być stosowane klucze RSA i DSA krótsze niż 1024 bity,
- w latach 2011-2030 powinny być używane algorytmy zapewniające min. 112 bitów - wycofany powinien zostać 3DES z dwoma kluczami (2TDEA), minimalna długość kluczy RSA i DSA to 2048 bitów,
- po roku 2030 powinny być używane algorytmy zapewniające min. 128 bitów - wycofany powinien być każdy wariant 3DES, minimalne długości kluczy RSA i DSA to 3072 bity

Zgodnie z zaleceniami SP 800-57 po aktualizacji w 2008 roku SHA-1 jako zapewniające odporność mniejszą niż 80 bitów nie powinno być stosowane w nowych implementacjach.

Funkcje uwierzytelniające

- MAC (*Message Authentication Code*)
 - Ochrona integralności
 - Ochrona autentyczności pochodzenia
 - Autentyczność zapewnia tajny klucz K
 - Nadawca generuje MAC i dołącza do wiadomości
 - $m, T := \text{MAC}(K, m)$ T – "authentication tag"
 - Odbiorca generuje MAC i porównuje z otrzymanym
 - $m, T =? \text{MAC}(K, m)$
 - Funkcja skrótu sparametryzowana tajnym kluczem

Termin polski - kod uwierzytelnienia wiadomości.

Funkcje MAC w praktyce

- HMAC (FIPS 198a)
 - $h(K \oplus a \parallel h(K \oplus b \parallel m))$
 - Najbardziej rozpowszechniony z SHA1
 - Zalecany z SHA-256
- CMAC (NIST SP 800-38A)
 - CBC-MAC, wykorzystuje AES-128, 192, 256
 - Zalecany z AES-256
- Realna złożoność $2^{n/2}$
 - Wymagane min. 128 bitów

Wymiana kluczy kryptograficznych

Skąd się biorą klucze?

- Liczby losowe w kryptografii
 - Krytyczne dla bezpieczeństwa
 - Klucze kryptograficzne, ciągi kluczowe, IV, nonce, identyfikatory...
- Źródła liczb losowych
 - Generatory pseudolosowe (PRNG)
 - Istotna jakość ciągu wyjściowego
 - Tylko PRNG zaprojektowane do celów kryptograficznych
 - Konieczne zasilenie (*seed*)
 - Generatory sprzętowe (*hardware RNG*)
 - Koszt, wydajność

52

PRNG – Pseudorandom number generator. Algorytm deterministyczny dający na wyjściu ciąg liczb mający własności statystyczne zbliżone do danych losowych.

Deterministyczność PRNG polega na generowaniu zawsze tego samego ciągu liczb, który w końcu zaczyna się powtarzać. Przykład (okres 5):

PRNG(1)=628, 98, 2780, 4, 790, 628, 98...

PRNG(2)=98, 2780, 4, 790, 628, 98, 2780...

PRNG(4)=790, 628, 98, 2780, 4, 790...

Kryptograficzne PRNG powinny charakteryzować się bardzo długim okresem powtarzalności.

Cechę pseudo-nieprzewidywalności osiągamy dzięki temu, że PRNG może generować ciąg od dowolnego kolejnego elementu – wskazujemy go parametrem zasilającym (*seed value*).

Entropia informacji

- Entropia informacji
 - suma wystąpień znaków w tekście w stosunku do wszystkich możliwych kombinacji
- Teksty rzeczywiste
 - Niska entropia (~4)
- Ciągi losowych bajtów
 - Wysoka entropia (~8)
- Dobry szyfr produkuje kryptogram o wysokiej entropii
 - Nawet jeśli entropia klucza i tekstu jawnego jest niska

53

Teksty rzeczywiste

- wszystkie znaki A-Z równie prawdopodobne
- niektóre występują znacznie częściej (EOI...)
- niektóre rzadko lub w ogóle
- entropia niska (rzędu 4,1, maksymalna 4,7 jeśli A-Z)

Ciągi losowych bajtów (0-255)

- maksymalna entropia 8,0 bitów/znak (bajt=8 bitów)
- dobre szyfry i generatory liczb losowych osiągają 7,6
- jeśli tekst naturalny w środowisku 8-bitowym to entropia relatywnie jeszcze mniejsza (4,1 vs 8,0)

Od hasła do klucza

- Niska złożoność haseł użytkowników
 - Średnio ok. 40 bitów entropii
- Techniki zwiększania złożoności
 - Modulowany skrót hasł do uwierzytelnienia
 - Ustawienie hasła: $salt, p := h(hasło \parallel salt)$
 - Logowanie: $h(hasło \parallel salt) = p?$
 - Iterowany skrót do szyfrowania
 - PKCS#5, Scrypt

54

"A Large-Scale Study of Web Password Habits",
Dinei Florêncio, Cormac Herley, Microsoft
Research

Colin Percival, Stronger Key Derivation via
Sequential Memory-Hard Functions, presented at
BSDCan'09, May 2009

PKCS #5 v2.1, "Password-Based Cryptography
Standard", RSA

NIST SP 800-118 Guide to Enterprise Password
Management

NIST SP 800-108 Recommendation for Key
Derivation Using Pseudorandom Functions

NIST SP 800-90 Recommendation for Random
Number Generation Using Deterministic Random
Bit Generators

Metody wymiany klucza

- Bezpieczne obliczenie wspólnego klucza
 - Algorytm Diffie-Hellman (1976)
 - Krzywe eliptyczne (ECDH)
- Kryptografia z kluczem publicznym
 - Algorytm RSA

55

Algorytm Diffiego-Hellmana jest historycznie pierwszym algorytmem kryptografii z kluczem publicznym. Umożliwia dwóm stronom komunikacji wspólne wyliczenie tajnej wartości. Proces ten polega na wymianie określonych liczb i wykonywaniu obliczeń na nich. Może się odbywać przez kanał publiczny. Ze względu na liczbę kroków wykonywanie DH jest preferowane w czasie rzeczywistym (on-line).

Algorytm RSA umożliwia szyfrowanie wiadomości kluczem publicznym i odszyfrowanie kluczem prywatnym. Każda z tych operacji odbywa się w jednym kroku, stąd nadaje się ona zarówno do pracy on-line jak i off-line (przesyłanie wiadomości).

Żaden z tych algorytmów nie gwarantuje autentyczności którejkolwiek ze stron!

NIST SP 800-57 Recommendation for Key Management
NISTP SP 800-56A, 800-56B

Wymiana klucza a uwierzytelnienie

- Atak *man-in-the-middle*
 - Niemożliwe rozwiązanie technikami kryptograficznymi
- Środki organizacyjne
 - Zaufana trzecia strona
 - PKI

Protokół kryptograficzny w praktyce

1) Przedstawienie się (*identification*)

- deklaracja tożsamości

2) Uwierzytelnienie (*authentication*)

- potwierdzenie zadeklarowanej tożsamości
 - podpis elektroniczny (RSA, DSA), lub
 - wspólne hasło (shared secret)
- W obu przypadkach konieczne dodatkowe mechanizmy organizacyjne!

57

Opisany schemat odpowiada sieciowym protokołom bezpieczeństwa takim jak IPSec czy SSL.

Tożsamością w tym przypadku jest arbitralnie wybrany identyfikator (adres IP, nazwa DNS, nazwa użytkownika w IPSec; nazwa domenowa w SSL).

Na jego podstawie strona odbierająca połączenie wybiera ustalone wcześniej dane uwierzytelniające (klucz publiczny, hasło).

Użycie podpisu cyfrowego do uwierzytelnienia wymaga zaangażowania PKI. Użycie hasła wymaga uzgodnienia tego hasła innym kanałem.

Protokół kryptograficzny w praktyce

3) Wymiana klucza (key exchange, key agreement)

- wymiana losowego klucza sesyjnego (RSA), lub
- ustalenie wspólnego klucza sesyjnego (DH)

4) Transmisja danych

5) Zarządzanie połączeniem

- Ustanawianie nowych kluczy co X godzin lub Y bajtów (*renegotiation*)

58

Jedno dwukierunkowe połączenie używa aż czterech symetrycznych kluczy kryptograficznych:

- jeden do szyfrowania (szyfr blokowy)
- jeden do ochrony integralności (HMAC)
- po jednym takim zestawie dla każdego kierunku transmisji (A->B, B->A)

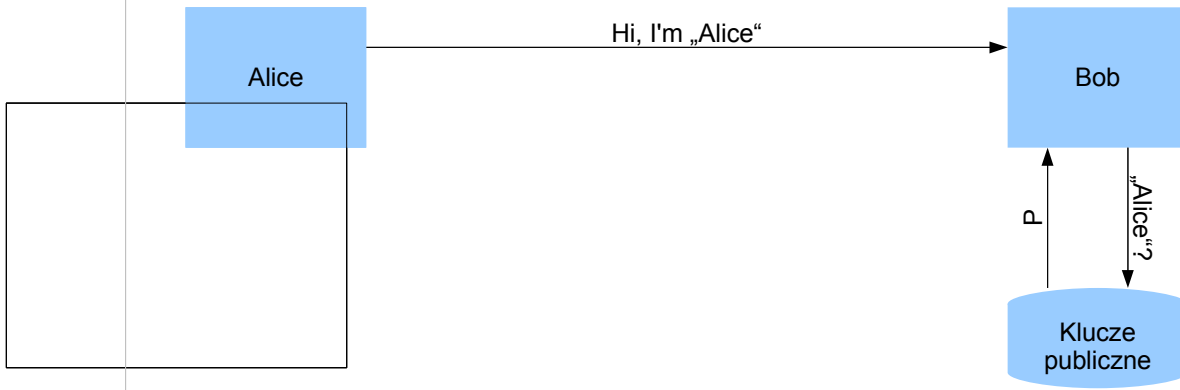
Wymiana wszystkich kluczy co ustalony interwał czasu chroni przed atakami kryptoanalitycznymi wymagającymi zebrania odpowiedniej ilości kryptogramów; co ustaloną ilość bajtów – przed atakami wykorzystującymi paradoks dnia urodzin lub meet-in-the-middle.

Jak widać, nawiązanie sesji wymaga dwukrotnego użycia RSA lub DH – po raz pierwszy do uwierzytelnienia, po raz drugi do wymiany klucza.

Identyfikacja



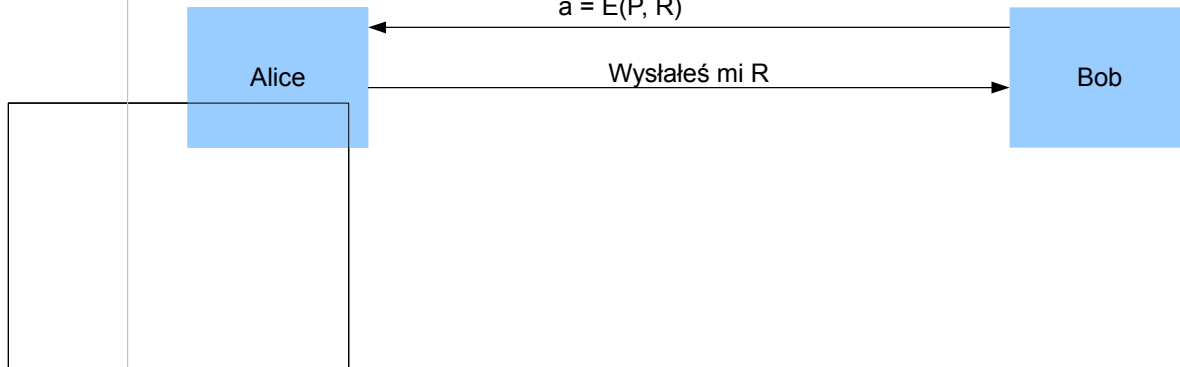
Identyfikacja



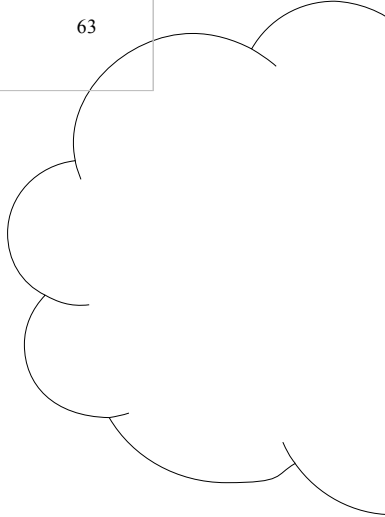
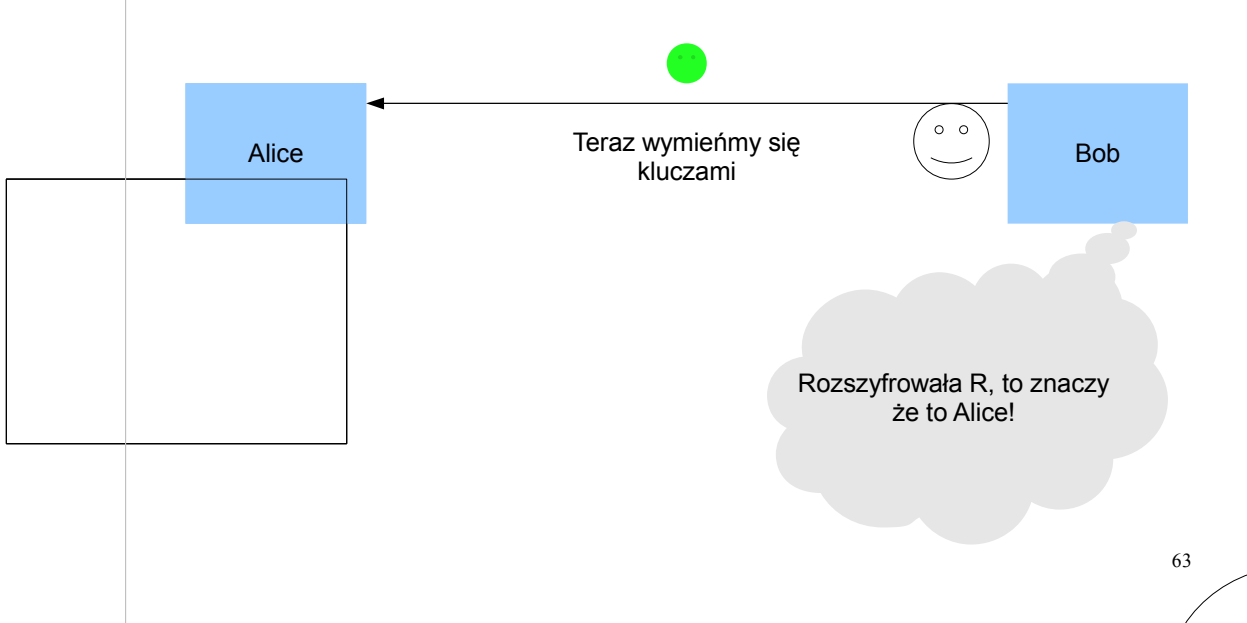
Uwierzytelnienie



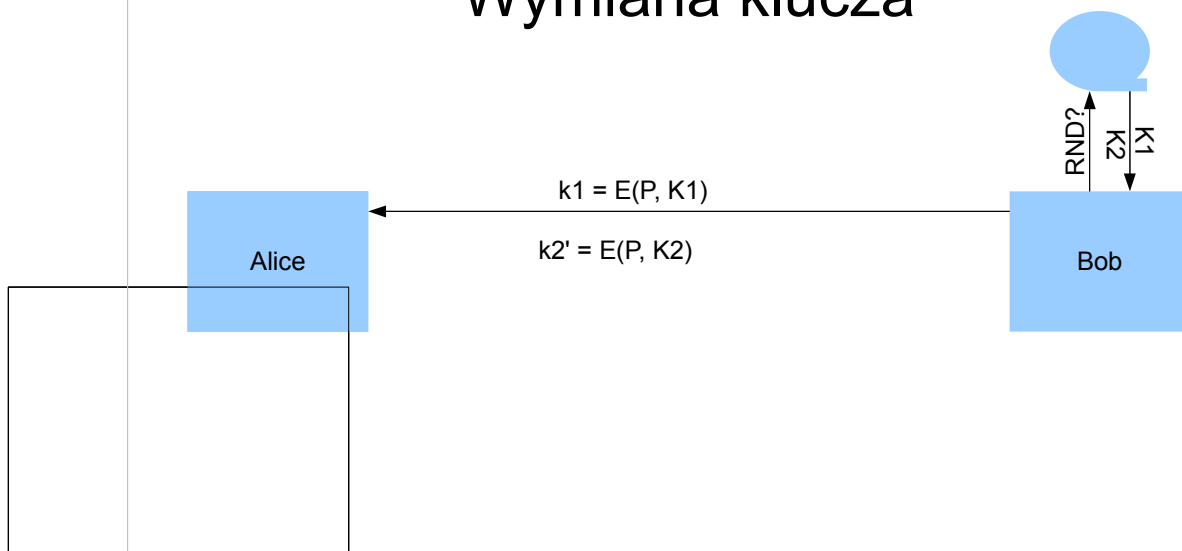
Uwierzytelnienie



Uwierzytelnienie



Wymiana klucza



Transmisja danych

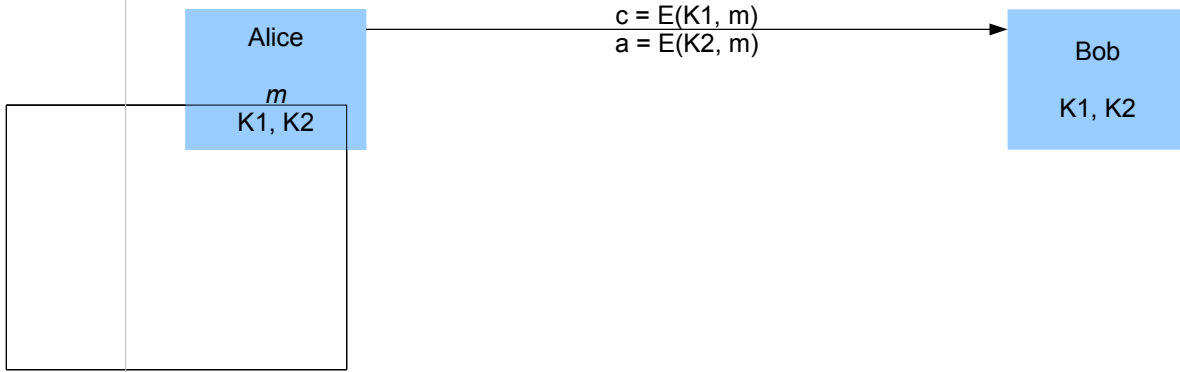
Alice

m
K1, K2

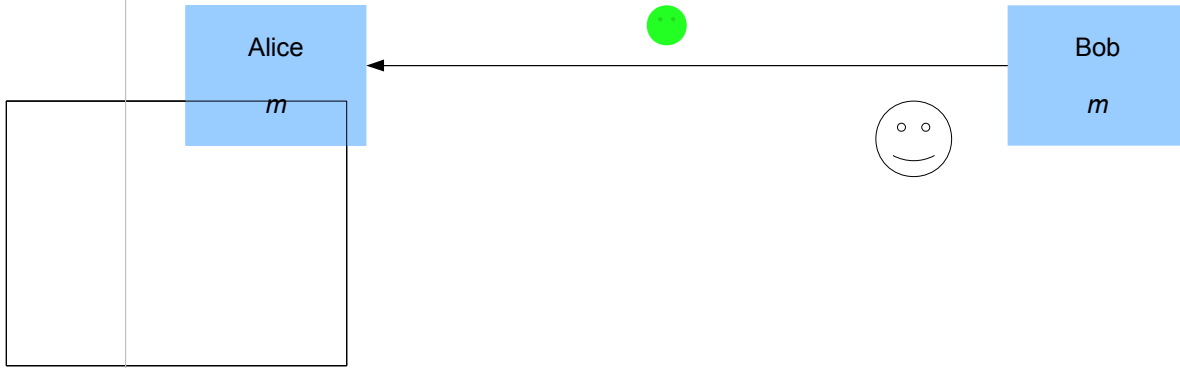
Bob

K1, K2

Transmisja danych



Transmisja danych



Kontakt z autorem:

pawel.krawczyk@hush.com

Tel. +48-602-776959

Prezentacja udostępniona na licencji Creative Commons BY-NC-SA
(„uznanie autorstwa”, „użycie niekomercyjne”, „na tych samych warunkach”)

<http://creativecommons.org/licenses/by-nc-sa/3.0/pl/>

68

Literatura:

<http://ipsec.pl/>

<http://securitystandard.pl/>

<http://csrc.nist.gov/>

Niels Ferguson, Bruce Schneier, t Tadayoshi Kohno
"Cryptography engineering"

Kontakt z autorem:

pawel.krawczyk@hush.com

Tel. +48-602-776959

Prezentacja udostępniona na licencji Creative Commons BY-NC-SA
(„uznanie autorstwa”, „użycie niekomercyjne”, „na tych samych warunkach”)

<http://creativecommons.org/licenses/by-nc-sa/3.0/pl/>

69

Literatura:

<http://ipsec.pl/>

<http://securitystandard.pl/>

<http://csrc.nist.gov/>

Niels Ferguson, Bruce Schneier, t Tadayoshi Kohno
"Cryptography engineering"