Pytania zamknięte

1. Pojęcie kryptografii bezkluczowej odnosi się do:
Algorytmów, w których do realizacji usługi związanej z bezpieczeństwem informacji nie jest wymagane istnienie przynajmniej jednego tajnego parametru

2. W kryptografii poufność doskonała w odniesieniu do algorytmów realizujących usługę poufności wymaga, aby:
Liczba różnych możliwych kluczy była równa co najmniej mocy przestrzeni wiadomości jawnych

3. Długość krytyczna(unicity distance) jest to:
Najmniejsza ilość zaszyfrowanego tekstu niezbędna do jednoznacznego określenia klucza szyfrowania

4. Kryptograficzny interfejs programowy (CAPI) dla „urządzeń" przechowywujących dane kryptograficzne i wykonujących operacje kryptograficzne, to przedmiot specyfikacji:
PKCS#11

5. Realizacja progowego podziału(4,4) sekretu reprezentowanego przez ciąg binarny liczący 4k bitów, gdzie k jest liczbą naturalną > 100, wymaga:
Arbitralnego powierzenia każdemu z udziałowców różnych fragmentów tego sekretu, z których każdy liczby k bitów

6. Proaktywne metody podziału sekretu dla schematu (m,n) umożliwiają:
Modyfikację wartości udziałów/cieni sekretu bez konieczności odtwarzania

7. Entropia idealnego ciągu losowego liczącego n bitów wynosi:
N bitów

8. Które z wymienionych poniżej zjawisk fizycznych nie może być wykorzystane do realizacji niedeterministycznego generatora liczb losowych (TRNG)?
Rejestracja dźwięku wydobywanego z rozstrojonego dętego instrumentu muzycznego (np. klarnetu)

9. Kryptograficznie bezpieczny generator binarnych ciągów pseudolosowych BBS wykorzystuje:
Problemy faktoryzacji liczby złożonej i wyznaczania pierwiastka kwadratowego w algebrach modularnych

10. Oparte na LFSR generatory „obcinajce" (shrinking generators) wykorzystywane do szybkiego generowania strumieni losowych bitów wykorzystują:
Wyjścia jednego (lub więcej) LFSR do „kluczowania" wyjść innego (lub innych) LFSR

11. Atak metodą dnia urodzin (birthday attack) na mechanizm kryptograficzne wykorzystujące funkcję skrótu skuteczny jest w przypadku:
Wszystkich funkcji skrótu

12. Algorytm funkcji skrótu SHA-3 (Keccak) strukturalnie oparty jest o schemat:
Gąbki(sponge) – „pochłania" skracanej wiadomości i „wyciskanie" skrótu

13. HMAC to:
Funkcja skrótu z tajnym kluczem

14. W grupie multiplikatywnej Z*27 elementem odwrotnym względem mnożenia modulo 27 do elementu o wartości 6 jest
element ten jest nieodwracalny w Z*27

15. Liczba elementów pierścienia (R, +, x) jest
Liczbą możliwych do utworzenia różnych par elementów zbioru R

16.Liczba elementów grupy multiplikatywnej ciała skończonego GF(pm), gdzie p jest liczbą pierwszą i m>1, wynosi:
pm-1

17.W „eliptycznych" wersjach algorytmów DH i DSA (odpowiednio ECDH, ECDSA) operację podnoszenia liczb do potęgi będącej liczbą całkowitą „zastąpiono"
Operacją mnożenia punktu na krzywej przez liczbę całkowitą

18. Która z poniższych tożsamości obowiązuje dla odwzorowań dwuliniowych (pairings):
$e([a]P,[b]Q) = e(P,Q)ab \; V \; a,b \in Z$

19. Pojęcie „lightweight cryptography" odnosi się do:
Mechanizmów kryptograficznych implementowanych w urządzeniach o ograniczonych zasobach i niewielkiej mocy obliczeniowej

20.Który z wymienionych ataków na rozwiązania implementujące kryptografię nie kwalifikuje się do grupy „side channel attacks"?
Atak ze znanym tekstem jawnym

21. Jaka struktura jest wykorzystywana w blockchain:
Drzewo Merkla

22. [zamknięte] W protokole o wiedzy zerowej opartym na problemie kolorowania 3-ma barwami wierzchołków planarnego grafu pewność, że podmiot uwierzytelniany zna "sekret" (czyli potrafi pokolorować w odpowiedni sposób wierzchołki) osiąga się po wskazaniu w kolejnych rundach (b):
wszystkich krawędzi grafu

23. [zamknięte] Technologia "blockchain" oprócz wskazania mechanizmów wiązania bloków danych o transakcjach, wymaga określenia:
protokołu konsensusu angażującego wiele podmiotów i umożliwiającego dołączenie kolejnego bloku do łańcucha

24. [zamknięte] Kiedy bit kwantowy (kubit) staje się "zwykłym" bitem (tzn, reprezentuje albo "0" albo "1"):
w chwili obserwacji dwupoziomowego stanu kwantowego

25. [zamknięte] W protokołach interaktywnych o wiedzy zerowej (sigma protocols) każda runda protokołu składa się z 4-ch faz. Pierwszą jest "zobowiązanie" (commitment), trzecią "odpowiedź" (response), zaś czwartą "weryfikacja" (verification). Wskazać niewymienioną powyżej fazę 2-gą:
"wyzwanie" (challenge)

26. [zamknięte] - co nie jest wykorzystywane w kryptografii kwantowej
(to z transformatą Fouriera)

27. [zamknięte] potęgowanie na krzywych eliptycznych
pewnie chodzi o pytanie z mnożeniem punktu na krzywej przez pewną stałą,

28. [zamknięte] czego wymaga technologia blockchain oprócz wiązania bloków
pewnie chodzi o pytanie z mechanizmem konsensusu;

29. [zamknięte] Które z wymienionych poniżej zjawisk fizycznych nie może być wykorzystane do realizacji niedeterministycznego generatora liczb losowych (TRNG)?
rejestracja dźwięku wydobywanego z rozstrojonego dętego instrumentu muzycznego (np. klarnetu)

30. [zamknięte] Kwantowy algorytm Grovera zmniejsza złożoność czasową procesu wyszukiwania elementu o określonej wartości w nieposortowanym zbiorze liczącym N elementów do klasy:

O(N^(½))

31. Schemat podpisu jednorazowego Lamporta wykorzystujący funkcję skrótu należy zaliczyć do

c) kryptografii asymetrycznej

Pytania otwarte

➔ **Na czym polegają proaktywne metody podziału sekretu i jaki jest sens ich stosowania?**

Polegają na dodaniu do wielomianu pierwotnego, wielomianu o pewnych właściwościach. Dodanie tego wielomianu stworzy nowy wielomian, który będzie chronił ten sam sekret. Aby odświeżyć cienie każdy udziałowiec wyznacza fragmenty odnowy sekretu i wysyła kanałami prywatnymi do innych udziałowców. W ten sposób w przypadku kompromitacji któregoś z cieni udziałowców możemy odświeżać cienie nie zmieniając sekretu.

➔ **Koncepcja "dostawcy usług kryptograficznych" (Cryptographic Service Provider) pojawia się w wielu specyfikacjach systemowych. Proszę wyjaśnić zwięźle logikę tej koncepcji oraz wskazać przykłady implementacji**

Dostawca usług kryptograficznych (CSP) to program obsługujący uwierzytelnianie, kodowanie i usługi szyfrowania udostępniane aplikacjom opartym na systemie Windows za pomocą interfejsu CryptoAPI firmy Microsoft. Poszczególni dostawcy CSP udostępniają różne implementacje interfejsu CryptoAPI. Niektórzy dostawcy oferują silniejsze algorytmy kryptograficzne, a inni korzystają ze składników sprzętowych, takich jak karty inteligentne.

Gdy użytkownik generuje żądanie nowego certyfikatu, informacje zawarte w tym żądaniu są najpierw wysyłane z programu żądającego do interfejsu CryptoAPI. Interfejs CryptoAPI przekazuje odpowiednie dane do dostawcy usług kryptograficznych, który został zainstalowany na komputerze użytkownika lub na urządzeniu dostępnym dla tego komputera. Jeśli dostawca CSP jest oparty na oprogramowaniu, generuje na danym komputerze klucz publiczny i klucz prywatny, które często nazywa się parą kluczy. Jeśli dostawca CSP jest oparty na sprzęcie, na przykład na karcie inteligentnej, zleca wygenerowanie pary kluczy określonemu urządzeniu.

Po wygenerowaniu kluczy dostawca CSP oparty na oprogramowaniu szyfruje, a następnie zabezpiecza klucz prywatny. Dostawca CSP korzystający z karty inteligentnej magazynuje klucz prywatny na karcie. Od tego momentu karta inteligentna kontroluje dostęp do klucza.

Klucz publiczny wraz z informacjami strony żądającej certyfikatu zostaje wysłany do urzędu certyfikacji. Gdy urząd certyfikacji sprawdzi żądanie certyfikatu zgodnie ze swoimi zasadami, używa własnego klucza prywatnego do utworzenia podpisu cyfrowego w certyfikacie, a następnie wystawia ten certyfikat stronie żądającej. Certyfikat od urzędu certyfikacji jest prezentowany stronie żądającej z możliwością zainstalowania go w odpowiednim magazynie certyfikatów na komputerze lub innym urządzeniu.

➔ **Krzywe eliptyczne dlaczego lepiej je stosować niż zwyczajne algorytmy takie jak RSA?**

ECC grupa technik kryptografii asymetrycznej, wykorzystująca jako podstawową technikę matematyczną krzywe eliptyczne. Oferuje bezpieczeństwo porównywalne do RSA przy znacznie krótszych kluczach. Ocenia się, że bezpieczeństwo klucza RSA o długości 1024 bitów jest równoważne bezpieczeństwu klucza ECC o długości 160 bitów. Z tego powodu ECC jest bardzo atrakcyjnym algorytmem w zastosowaniach, które wymagają bardzo wysokiej wydajności szyfrowania asymetrycznego (algorytm RSA jest stosunkowo wolny) lub oferują bardzo ograniczone środowisko obliczeniowe (jak karty mikroprocesorowe). ECC wykorzystuje działania matematyczne krzywych eliptycznych w ciałach, na liczbach całkowitych i w oparciu o duże liczby pierwsze.

➔ Na czym polega kleptografia i jak się przed nią chronić?

Polega na takim preparowaniu sprzętowych (lub opartych tylko na oprogramowaniu) modułów kryptograficznych, by umożliwić przechwycenie przez „producenta" modułu sekretów podmiotów je wykorzystujących, które teoretycznie powinny pozostawać bezpieczne wewnątrz modułu podczas realizowania usług kryptograficznych. Wykrycie ataku kleptograficznego wyłącznie na podstawie obserwacji danych wejściowych i wyjściowych z urządzenia jest praktycznie niemożliwe. Atak polega wyłącznie na biernym podsłuchiwaniu sygnałów (parametrów) wymienianych między modułem a światem zewnętrznym, a do jego przeprowadzenia niezbędne jest „zainstalowanie" w module (przed powierzeniem go właściwemu użytkownikowi) dodatkowego tajnego sekretu dostawcy modułu, oraz odpowiedniego spreparowania oprogramowania samego modułu

WYTŁUMACZENIE JAK DLA DEBILA:

Kleptografia to tak naprawdę celowe osłabienie systemu kryptograficznego, żeby łatwiej można było go zaatakować, ale żeby nie był istotnie słabszy, jeżeli atakującym jest osoba trzecia. (Źródło -> Wikipedia)

OCHRONA -- Chronić się przez: używanie open-sourcowego oprogramowania, unikać systemów opartych na zamkniętym oprogramowaniu.Używanie kanałów ukrytych/podprogowych

➔ Czym jest kubit i na czym polega zjawisko splątania (entanglement) kubitów?

Bitem kwantowym (kubitem) jest dwupoziomowy układ kwantowy. Stanami bazowymi są dwa stany, oznaczane zwyczajowo jako |0> i |1>. Stan splątany nie zależy od wyboru baz rozpatrywanych stanów kwantowych
Dla debila: Kubit – najmniejsza i niepodzielna jednostka informacji kwantowej. Z fizycznego punktu widzenia, kubit jest kwantowomechanicznym układem opisanym dwuwymiarową przestrzenią Hilberta – w związku z tym, różni się od klasycznego bitu tym, że może znajdować się w dowolnej superpozycji dwóch stanów kwantowych.

kubity(pairing) jeśli są singletem i oddalimy je od siebie i zmierzymy spin jednej to dostniemy info o spinie drugiej.

Jeśli dwie cząstki tworzące singlet oddalimy od siebie i zmierzymy spin jednej z nich, dostaniemy informację także o spinie tej drugiej. Ponieważ stan każdej z nich przed pomiarem jest nieokreślony, a z mechaniki kwantowej nie wynika żadne opóźnienie, więc Einstein, Podolski i Rosen doszli do wniosku, że oznacza to natychmiastowe oddziaływanie z nieskończoną prędkością, co jest sprzeczne z teorią względności. Einstein nazwał to zjawisko „upiornym działaniem na odległość".

Stan splątany - przeciwieństwo stanu rozkładalnego

KOMENTARZ : W znanych nam do tej pory układach scalonych temu układowi binarnemu odpowiadają dwa rodzaje wartości napięcia na tranzystorach. W modelu kwantowym zamiast dwóch możliwości pojawiają się skomplikowane połączenia, zwane splątaniami. Zamiast albo zerem, albo jedynką, kubit jest jednocześnie zerem lub jedynką. Dwa kubity to już cztery splątane ze sobą wartości (00, 01, 10, 11), trzy kubity – osiem (000, 001, 010, 100, 101, 110, 011, 111) i tak dalej. Stanów tych nie można traktować jako niezależnych, dlatego mówi się o nich, że są ze sobą splątane.

➔ Podać przykład realizacji niedeterministycznych generatorów losowych ciągów binarnych (TRNG)? Jaka jest ich podstawowa zaleta w porównaniu z generatorami deterministycznymi (PRNG)

Zasadą działania niedeterministycznych NRNG (TRNG) jest pozyskiwanie ciągów binarnych z nieprzewidywalnego i nie kontrolowanego przez człowieka fizycznego zjawiska traktowanego jako źródło sygnałów („wiadomości") o określonej (najlepiej maksymalnej) entropii:

- Termiczne wahania wartości rezystancji
- Szumy termiczne diod

- Szumy atmosferyczne
- Licznik Geigera
- Dźwięk mikrofonu
- Sygnał wideo z kamery
- Turbulencje powietrza w zamkniętym napędzie dyskowym, które powodują przypadkowe wahania czasu oczekiwania na odczyt sektora

Podstawową zaletą TRNG nad PRNG jest brak okresowości generowanych ciągów

➔ **Czy zdecydował(a)ł(a)by się Pani/Pan na korzystanie z kryptowalut? Odpowiedź(tak albo nie) proszę uzasadnić odnosząc się do cech kryptograficznych i systemowych takich rozwiązań**

Tak, ponieważ pozwalają na transakcję bez zaufanej trzeciej strony (banku, państwa). (Pamiętaj że będzie potrzebna trzecia strona jeśli chcesz portfel online, w przeciwnym razie ponad 250gb portfela offline) Transakcje bazują na dowodach kryptograficznych zamiast zaufaniu. Transakcje są zrealizowane w taki sposób, że odwrócenie ich i wprowadzenie fałszywych transakcji jest problemem trudnym obliczeniowo.

KOMENTARZ: Z perspektywy kryptograficznej jest spoko, z perspektywy środowiskowej już niekoniecznie. Emisja CO2, zużycie prądu, wykupywanie kart graficznych. Pozyskanie 51% mocy obliczeniowej może skutkować selfish mining attackiem/

Selfishminingattack-zamiast publikować wydobyty blok, kopacz zatrzymuje go dla siebie i liczy następny, podczas gdy konkurenci tracą moc obliczeniową na obliczanie starego bloku; po pewnym czasie publikuje on dłuższy łańcuch, który staje się obowiązujący, atak działa wtedy, gdy kontroluje się co najmniej około 1/3 mocy obliczeniowej sieci.

➔ **Co „kryje się" za pojęciem odwzorowań dwuliniowych (pairings)?**

Parowanie jest definiowane jako bilinearna mapa pomiędzy elementami dwóch skończonych, cyklicznych i addytywnych grup G1 i G2 do trzeciej skończonej cyklicznej grupy G3 zdefiniowanej multiplikatywnie.
e: G1 x G2 -> G3 Musi spełnić 3 warunki: dwuliniowości, niezdegenerowania, obliczalności.

```
public static void pairingtests() {
    //a - inicjalizacja
    int size = 160;
    //int size = 256;
    //int size = 384;
    //int size = 512;
    //int size = 638;
    //inicjalizacja odwzorowania typu 3
    final Pairing pair3 = AtePairingOverBarretoNaehrigCurveFactory.getPairing(PairingTypes.TYPE_3, size);
    EllipticCurve curve1 = pair3.getGroup1();
    EllipticCurve curve2 = pair3.getGroup2();
    //b - liczby losowe
    final SecureRandom random = SecurityStrength.getSecureRandom(
            SecurityStrength.getSecurityStrength(curve1.getField().getFieldSize()));
    //tworzenie liczby losowej
    BigInteger s_TA = new BigInteger(size - 1, random);
    //c - pobieranie generatora dla danej krzywej
    ECPoint P = curve1.getGenerator();
    ECPoint Q = curve2.getGenerator();
    //mnożenie punktu przez skalar
    //jak potrzebujemy losowy punkt na krzywej, to losujemy punkt i mnożymy przez skalar
    ECPoint P_0 = P.multiplyPoint(s_TA);
    ECPoint Q_0 = Q.multiplyPoint(s_TA);
    //obliczanie odwzorowania
    GenericFieldElement e1 = pair3.pair(P, Q_0);
    GenericFieldElement e2 = pair3.pair(P_0, Q);
    System.out.println(e1);
    System.out.println(e2);
    if (e1.equals(e2)) {
        System.out.println("Wszystko działa jak powinno :)");
    }
```

➔ **Proszę omówić schemat podpisu jednorazowego Lamporta-Diffiego i wskazać jego podstawową wadę**

Podstawa dla systemów opartych na funkcjach skrótu. Podpis jednorazowy, polega na wygenerowaniu pary kluczy dla podpisania wiadomości k-bitowej. Funkcja jednokierunkowa to k-bitowa funkcja skrótu h(.). Podpisujący generuje k par losowych ciągów k-bitowych. Klucz prywatny - zestaw 2k^2-bitowy , każdy z k-bitowych ciągów losowych indeksuje się pozycją i wartością binarną. Klucz publiczny - też 2k^2-bitowy zestaw , jego elementy indeksowane pozycją i wartością binarną ciągu. Podpis - k-elementowy ciąg k-bitowych wartości. Weryfikacja - jest to podpis z załącznikiem, weryfikator zna wiadomość, podpis i klucz publiczny. Sprawdzamy każdy bit wiadomości. Wada: Przy podpisywaniu więcej niż jednej wiadomości stosujemy "chaining" i umieszczamy nowy klucz w podpisanej wiadomości dla następnej wiadomości. Podpisy rosną liniowo wraz ze wzrostem ich liczby tworzonej przez jednego podpisującego.

➔ **Dlaczego liniowe rejestry przesuwające są wykorzystywane do fizycznej implementacji algorytmów kryptograficznych? Proszę wskazać systemy/rozwiązania techniczne, w których można je napotkać w tej roli.**

- LFSR są łatwe do implementacji sprzętowej;
- mogą generować ciągi binarne o dużych okresach;
- mogą generować ciągi binarne o dobrych z punktu widzenia kryptografii cechach statystycznych;
- dzięki swojej strukturze mogą być skutecznie i łatwo analizowane za pomocą metod algebraicznych.

Przykłady:

- Nieliniowe generatory kombinacyjne (np. Geffe'go)
- Generatory ze sterowanym wejściem taktującym (Stop and Go)
- Generatory z kluczowanym wyjściem ("shrinking generators")
- Algorytm szyfrujący A5 stosowany w GSM
- Algorytm CRYPTO1 stosowany w płatnościach zbliżeniowych z układem MIFARE®STANDARD

➔ **W jaki sposób określa się logarytm dyskretny w addytywnej grupie punktów na krzywej eliptycznej?**

$Log_n(x)$

gdzie n jest całkowitoliczbowe

Wprowadza się także pojęcie logarytmu dyskretnego na krzywej eliptycznej (*ECDL*):

jeżeli $R = kP$, to $log_P R = k$.

➔ **Proszę wyjaśnić na czym polega "paradoks dnia urodzin" w odniesieniu do funkcji skrótu i jakie ma on znaczenie dla bezpieczeństwa mechanizmów /algorytmów/protokołów kryptograficznych**

Związek paradoksu dnia urodzin z funkcją skrótu jest taki, że znalezienie pary wiadomości o tym samym skrócie (kolizja) jest równe znalezieniu pary osobników urodzonych w tym samym dniu roku liczącego 2^n dni.

Paradoks dnia urodzin jest podstawą działania ataku urodzinowego, którego procedura jest następująca:

1. należy wygenerować $2^{n/2}$ fałszywych wiadomości $m_i (i = 1,2, \ldots, 2^{n/2})$
2. należy wygenerować $2^{n/2}$ fałszywych podpisów $s_i (i = 1,2, \ldots, 2^{n/2})$
3. utworzyć listę $E_e(si)$ oraz listę h(mi)

4. znaleźć parę (j, k) taką, że Ee(sj) = h(mk)

gdzie n to wartość funkcji skrótu w bitach.

WYTŁUMACZENIE JAK DLA DEBILA:

Generujemy sobie $2^{n/2}$ fałszywych wiadomości oraz fałszywych podpisów. Następnie robimy sobie listę zaszyfrowanych fałszywych podpisów i listę skrótów fałszywych wiadomości. Po utworzeniu tych list szukamy kolizji między zaszyfrowanymi fałszywymi podpisami oraz skrótami wiadomości. Jak się trafi to atak jest udany. (zweryfikujcie czy git)

➔ **Proszę wyjaśnić pojęcia grupy, pierścienia i ciała .**

Grupą (G, o) jest zbiór G z operacją binarną o. Spełnia ona trzy aksjomaty:

1. Dla każdego a,b,c należącego do zbioru G musi zachodzić łączność tj. a o (b o c) = (a o b) o c
2. Istnieje element e należący do zbioru G dla każdego a należącego do zbioru g który jest tożsamościowy tj. a o e = e o a = a
3. Dla każdego a należącego do zbioru G istnieje element odwrotny a^{-1} należący do zbioru G tj. a o a^{-1} = a^{-1} o a = e

Jakaś operacja binarna musi być :
łączna → a + (b + c) = (a + b) + c
tożsamościowa → mamy element który nic nie zmienia 1 + 0 = 0 + 1 = 1
posiada element odwrotny należący do zbioru.

Jeżeli wartość G (moc zbioru) jest skończona, to grupa G jest grupą skończoną, liczba elementów grupy jest rzędem grupy

Pierścień (R, +, x) tworzy zbiór R z dwoma operacjami binarnymi, arbitralnie oznaczonymi jako + (dodawanie) oraz x (mnożenie).

Własności pierścienia (o ile mogę to tak nazwać, ewentualnie aksjomaty):

(R, +) jest grupą przemienną z elementem tożsamościowym oznaczonym jako 0 (element zerowy, neutralny):

- Dla każdego a, b, c należącego do zbioru R zachodzi łączność tj. a x (b x c) = (a x b) x c
- Istnieje taki element e należący do zbioru R, który jest tożsamościowy tj. (e != 0) oraz (dla każdego a należącego do zbioru R: a x e = e x a = a)

Operacja x jest rozdzielna względem operacji +:

- dla każdego a,b,c należącego do zbioru R zachodzi zależność: a x (b + c) = (a x b) + (a x c)
- dla każdego a,b,c należącego do zbioru R zachodzi zależność: (b + c) x a = (b x a) + (c x a)

Pierścieniem nazywamy algebrę (R,+,·), gdzie (R,+) jest grupą abelową,

Ciałem F jest pierścień przemienny, którego wszystkie niezerowe elementy są elementami odwracalnymi. Gdzie pierścień przemienny to taki pierścień, który spełnia aksjomaty normalnego pierścienia oraz dla każdego a, b należącego do zbioru R zachodzi zależność a x b = b x a.

➔ **Jaki problem w czasie wielomianowym można rozwiązać za pomocą kwantowego algorytmu Petera Shora? Jakie to ma znaczenie dla kryptoanalizy?**

Za pomocą kwantowego algorytmu Petera Shora można rozwiązać w czasie wielomianowym problem faktoryzacji (rozkładu liczb złożonych) i wyznaczania logarytmu dyskretnego. W kryptoanalizie ma to duże znaczenie przez wzgląd na bezpieczeństwo. Wiele algorytmów kryptografii asymetrycznej bazuje na tych problemach np. RSA, ElGamal/DSA, protokół DH (i liczbowo, i na krzywych). Gdy będzie możliwe zastosowanie w praktyce algorytmu Petera Shora, algorytmy te nie będą już bezpieczne.

KOMENTARZ: Algorytm ten stanowi teoretyczne zagrożenie dla powszechnie używanego w internecie kryptosystemu RSA. Klucz publiczny w RSA jest iloczynem dwóch dużych liczb pierwszych. Możliwość efektywnego odtworzenia tych liczb na podstawie klucza publicznego pozwalałaby poznać klucz prywatny i tym samym złamać cały szyfr.

➔ **Akronim AEAD zazwyczaj pojawia się w kontekście "kryptografii wagi lekkiej" (lightweight cryptography). Do jakiej usługi kryptograficznej się odnosi?**

Szyfrowanie uwierzytelnione i uwierzytelnianie za pomocą powiązanych danych to formy szyfrowania, które jednocześnie zapewniają poufność i autentyczność danych

KOMENTARZ : AEAD - message-based encryption functions (funkcje szyfrowania oparte na wiadomościach)

➔ **Jaki jest związek pomiędzy pojęciami "blockchain" i "distributed ledgers technology"? Proszę wskazać podstawowe cechy tych rozwiązań.**

Blockchain wymaga bezpieczeństwa. Rejestr powinien być odporny na naruszenia i spójny pośród uczestników. DLT - distributed ledgers technology, wykrywa naruszenie, gdy węzeł może porównać wartość skrótu z innymi węzłami w celu wykrycia takiego naruszenia. Pojęcie to jest też związane ze zwięzłym nieinteraktywnym dowodem o wiedzy zerowej.

➔ **Proszę wskazać co najmniej trzy "rodziny" algorytmów kryptograficznych, w stosunku do których zakłada się, że mogą skutecznie chronić przed "siłą" ataku za pomocą komputerów kwantowych.**

- Kryptografia symetryczna (o odpowiednio dużych rozmiarach kluczy);
- Hash-based cryptography (funkcje jednokierunkowe bez „trap-door");
- Lattice-based cryptography (wykorzystanie problemu poszukiwania najbliższego wektora w kracie, np. systemy NTRU);
- Code-based cryptography (systemy oparte na kodach korygujących błędy, np.: schemat podpisu McEliece'a z kodem Goppy);
- Multivariate quadratic equations cryptography (równania kwadratowe wielu zmiennych, np. „oil & vinegar" HFEv system);

- Supersingular elliptic curve isogeny cryptography (wykorzystanie izogenii supersingularnych/super-osobliwych krzywych eliptycznych).

➔ **Jaki jest związek szyfru jednorazowego "one-time pad" Vernama z bezpieczeństwem systemów szyfrujących?**

Jaką podstawową cechę musi mieć ten szyfr?

Szyfr jednorazowy "one-time pad" Vernama jest ściśle powiązany z zagadnieniem długości krytycznej. W Vernamie długość klucza szyfrującego jest równa długości tekstu jawnego. Długość krytyczna jest to najmniejsza wartość N dla której $H_c(K) = 0$, a więc najmniejsza ilość zaszyfrowanego tekstu niezbędna do jednoznacznego określenia klucza szyfrowania.
Cecha szyfru: długość klucza szyfrującego jest równa długości tekstu jawnego. Hc(K) - entropia przestrzeni kluczy (jakby coś)

➔ **NTRU - (N-truncated polynomials) - bazowe operacje**

NTRU - kryptografia asymetryczna wykorzystujący wielomiany obcięte

generowanie pary kluczy -> szyfrowanie -> deszyfrowanie -> uzasadnienie poprawności deszyfrowania

\/\/\/\/\/\/ **podstawowe operacje**

dodawanie wielomianów - tak samo jak w pierścieniu wielomianów

mnożenie wielomianów - nie kończy się po obliczeniu współczynników wielomianu stopnia 2(N-1), lecz po wykonaniu mnożenia składniki wielomianu postaci $a_{N+k}x^{N+k}$, gdzie $0 \le k \le N-2$, są zastępowane przez $a_{N+k}x^k$, a następnie współczynniki przy składnikach o tych samych potęgach zmiennej x są sumowane.

$$f(x) = a_{N-1}x^{N-1} + \ldots + a_2x^2 + a_1x + a_0$$
$$g(x) = b_{N-1}x^{N-1} + \ldots + b_2x^2 + b_1x + b_0$$

$$h(x) = f(x) + g(x) = (a_{N-1} + b_{N-1})x^{N-1} + \ldots + (a_2 + b_2)x^2 + (a_1 + b_1)x + (a_0 + b_0).$$

Operacja mnożenia wielomianów nie kończy się po obliczeniu współczynników wielomianu stopnia $2(N-1)$, lecz po wykonaniu mnożenia składniki wielomianu postaci $a_{N+k}x^{N+k}$, gdzie $0 \le k \le N-2$, są zastępowane przez $a_{N+k}x^k$, a następnie współczynniki przy składnikach o tych samych potęgach zmiennej x są sumowane.

$$R = Z \quad N = 3 \quad f(x) = 3x^2 - x + 2 \quad g(x) = -x^2 + 2x + 1$$

$$f(x) + g(x) = (3-1)x^2 + (-1+2)x + (2+1) = 2x^2 + x + 3$$

$$f(x)g(x) = (3x^2 - x + 2)(-x^2 + 2x + 1) = -3x^4 + 7x^3 - x^2 + 3x + 2 =$$
$$= \underline{-3x + 7} - x^2 + 3x + 2 = -x^2 + 9$$

➔ **NTRU:(szukamy odp)**

NTRU - kryptografia asymetryczna wykorzystujący wielomiany obcięte

podstawowe operacje: generowanie pary kluczy -> szyfrowanie -> deszyfrowanie -> uzasadnienie poprawności deszyfrowania

podstawowe operacje chyba o te chodzi tutaj: operacja dodawania wielomianów zdefiniowana jest tak, jak w pierścieniu wielomianów R[x]

Podkreśla się następujące „przewagi" systemów NTRU w stosunku do systemów opartych na pierścieniach liczb całkowitych Zn (np. RSA), czy systemach opartych na krzywych eliptycznych (ECC):

porównywalne bezpieczeństwo przy 100-krotnie szybszych obliczeniach;

krótkie, łatwo generowane klucze;

małe wymagania zasobów pamięci i mocy obliczeniowej;

większa elastyczność przy wyborze parametrów stosownie do potrzeb użytkownika;

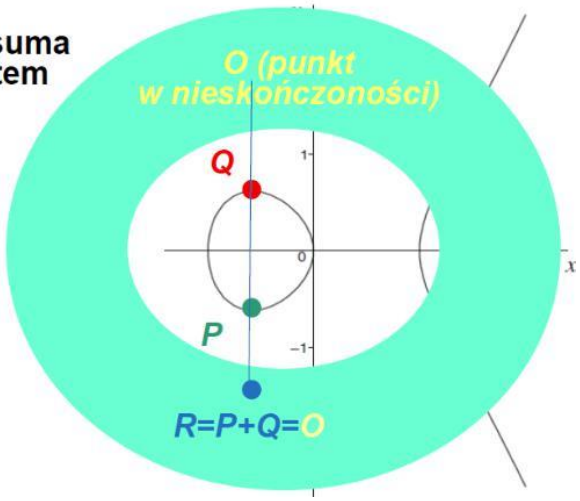
nieznane kwantowe algorytmy krypto analityczne.

➔ **Jaka jest klasyczna interpretacja graficzna dodawania punktów na krzywej eliptycznej oraz punktu neutralnego ze względu na dodawania ("zera")?**

[nie wiadomo] Jeśli dodamy punkt zerowy do punktu, to nowy punkt nie zmienia swojego położenia, ponieważ punkt O jest elementem tożsamościowym (neutralnym).

Jeżeli $x_P = x_Q$ oraz $y_P = -y_Q$ (a zatem wtedy, gdy $P = -Q$), to suma tych punktów musi być elementem neutralnym O, czyli "punktem w nieskończoności".

Przykład dla $K = R$



➔ **Jaka interpretacja pojęcia entropii w kryptologii?**

Jest to miara ilości informacji pozyskanej w wyniku obserwacji zmiennej losowej X:

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i \quad \text{(w bitach)}$$

Liczba bitów informacji, którą trzeba pozyskać, aby ją ujawnić. Entropia mierzy nieokreśloność wiadomości. Każda dodatkowo pozyskana informacja zmniejsza entropię.

➔ **Jaki problem można rozwiązać za pomocą kwantowego algorytmu Lova Grovera?**

Jest to algorytm kwantowy wyszukujący określoną wartość w nieposortowanym zbiorze liczącym N elementów w czasie O(pierwiastek z N albo N^½), gdzie klasycznie jest średnio N/2. Np. przeszukiwanie bazy danych. Zagrożenie dla kryptografii symetrycznej i asymetrycznej.

➔ **Rząd addytywnej grupy punktów na krzywej eliptycznej jest pewną liczbą pierwszą. Ile generatorów ma taka grupa (dla wszystkich grupa prawdziwe jest twierdzenie Lagrange'a)?**

➔ **Jaki powinien być dobry generator losowy i trzy mechanizmy kryptograficzne gdzie są wykorzystywane generatory.**

Generator losowych bitów to urządzenie lub algorytm, które generują ciąg statystycznie niezależnych i nieobciążonych bitów. Każdy bit idealnego ciągu losowego jest nieprzewidywalny i nieobciążony;

Jego wartość jest niezależna od wartości innych bitów tego ciągu, zaś prawdopodobieństwa tego, iż ta wartość wynosi 0 albo 1, są identyczne.

Entropia idealnego ciągu losowego liczącego n bitów wynosi n bitów.

Ponadto idealny ciąg losowy jest nieokresowy.

Przykłady:

- Generowanie pary kluczy kryptografii asymetrycznej (ciąg losowy musi zazwyczaj spełniać dodatkowe warunki wynikające z logiki algorytmu(;
- Generowanie jednorazowych kluczy sesyjnych kryptografii symetrycznej (jednostronne lub wielostronne, jak np. D-H);
- Generowanie losowych wyzwań w protokołach uwierzytelniania (tzw. nonce);
- Generowanie strumienia klucza dla szyfratorów strumieniowych;
- Generowanie losowych „wtrętów" (salts, seeds) w algorytmach kryptografii klucza publicznego w celu uniknięcia powtarzalności przetwarzania tych samych danych tym samym kluczem (prywatnym albo publicznym);

➔ **Jakie czynniki bierze się pod uwagę rozważając "wagę" rozwiązań sprzętowo-programowych kwalifikowanych jako "lightweight cryptography"?**

Waga oprogramowania – konsumpcja zasobów czasowych. Liczba cykli koniecznych do przetworzenia bajta danych oraz liczba cykli do przygotowania właściwej obsługi danych.

Waga sprzętu – liczba bramek logicznych niezbędnych do implementacji algorytmów.

Pobór mocy – urządzenia wagi lekkiej mają ograniczoną ilość energii.
(?)Kompromis między wymaganiami programowymi i sprzętowymi
(?)Zmniejszenie odporności na ataki typu „Side-Channel Attack"

➔ **Jakie jest uzasadnienie stosowania "eliptycznych odpowiedników" takich algorytmów jak RSA, DSA, czy protokół DH? Na czym polega przeprowadzka z obliczeń na liczbach całkowitych na "manipulowanie" punktami na krzywej eliptycznej?**

Polega na tym, że zamiast potęgowania przez liczbę całkowitą, mnożymy punkt na krzywej przez pewną stałą całkowitą

➔ **Grupowe podpisy cyfrowe**

- podpisy mogą składać wyłącznie członkowie grupy
- weryfikator może stwierdzić ważność podpisu, lecz nie może ustalić tożsamości członka składającego podpis
- w przypadku sporu musi istnieć możliwość ujawnienia tożsamości podmiotu składającego podpis
- Grupy są zwykle statyczne(określona ilość członków od początku) lub dynamiczne(co wpływa na mechanizmy stosowane w systemie).

➔ **Czym jest moduł kryptograficzny i jakie warunki musi spełniać i jakie funkcje może pełnić.**

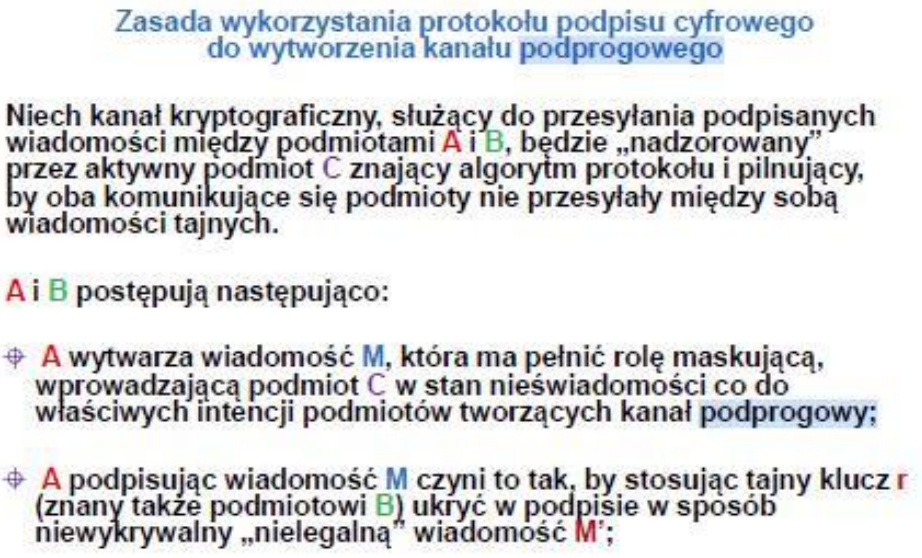
Jest to wydzielony moduł sprzętowy (hardware) i/lub programowy (software) zawierający zaimplementowane mechanizmy, procesy, algorytmy kryptograficzne, umieszczony w obszarze kryptograficznym

Funkcje:

- przechowuje klucze kryptograficzne i materiał kluczowy
- szyfruje i deszyfruje dane
- oblicza wartości podpisów cyfrowych, weryfikacja
- tworzy tokeny dla protokołów uwierzytelniania
- bezpieczne zarządzanie kluczami
- odtwarzanie sekretów

➔ **Pytanie o metodę podprogowa. ?? przykład trzeba było podać**

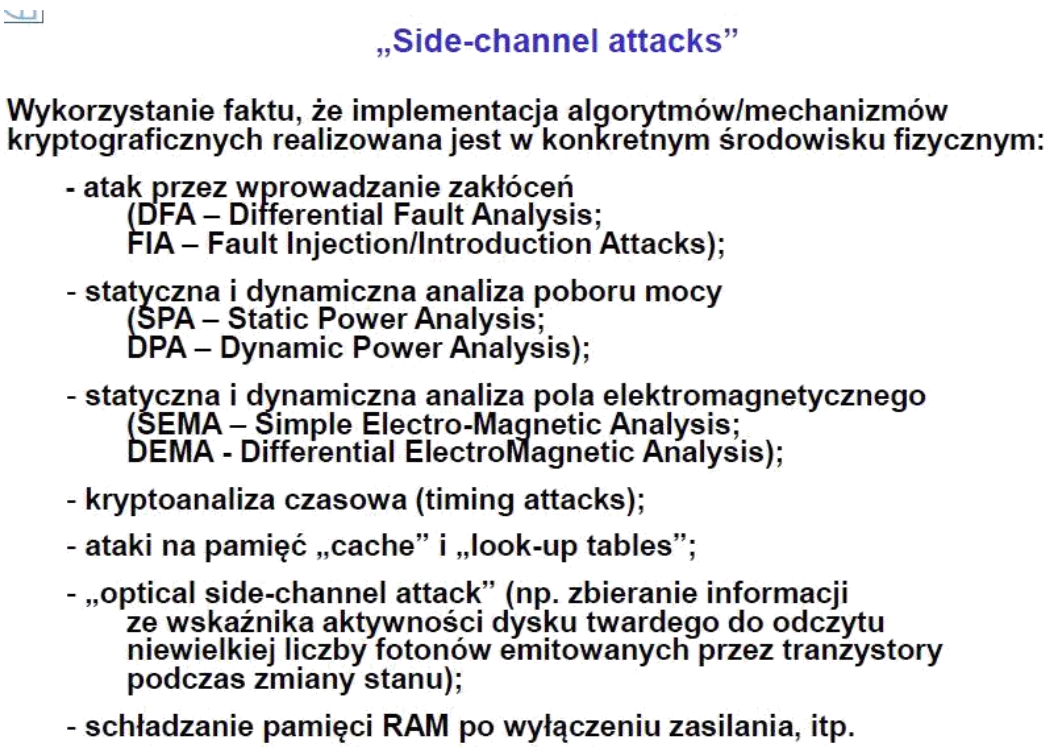
Większość systemów kryptograficznych umożliwia, dzięki nadmiarowości informacji przekazywanej w kryptogramach lub podpisach cyfrowych, przesyłanie (w ramach formalnie poprawnego protokołu, który może być weryfikowany przez Zaufaną Trzecią Stronę) dodatkowych informacji - bez ujawniania ich obecności stronie weryfikującej. W przypadku, gdy do tego celu są wykorzystywane schematy podpisów cyfrowych - mówi się o tzw. kanałach podprogowych (subliminal channels), zaś generalnie taki sposób wykorzystywania kanału/protokołu kryptograficznego określany jest mianem nadużycia (abuse).



Metody progowe podziału sekretu – powinny spełniać dwa warunki:

- Znajomość m<=n cieni sekretu
- Odtworzenie sekretu na podstawie znajomości i<m cieni jest problemem trudnym obliczeniowo.

SIDE CHANNEL ATTACKS



➔ **Dlaczego protokoły o wiedzy zerowej**

Bo nie ujawnia się ani cząstki klucza przy dowodzeniu jego posiadania. A dodatkowe odpowiedzi na “wyzwania” zmniejszają prawdopodobieństwo oszukiwania przez podmiot. Przykładowo protokół z labiryntem, kolorowania grafu.
Strona uwierzytelniana A przekonuje stronę uwierzytelniającą B, ze jest w posiadaniu sekretu, ale protokół jest skonstruowany tak by A nie musiał ujawnić ani cząstki sekretu zaś jego prawidłowe odpowiedzi zmniejszały prawdopodobieństwo oszukania przez A.

➔ Algorytm RSA – generowanie pary kluczy, szyfrowanie i deszyfracja:

Generowanie kluczy [edytuj | edytuj kod]

W celu wygenerowania pary kluczy (prywatnego i publicznego) należy posłużyć się algorytmem:

- Wybieramy losowo dwie duże liczby pierwsze *p* i *q* (najlepiej w taki sposób, aby obie miały zbliżoną długość w bitach, ale jednocześnie były od siebie odległe wartościami – istnieją *lepsze* mechanizmy faktoryzacji, jeżeli liczba ma dzielnik o wartości *bliskiej* $\sqrt{n}$).
- Obliczamy wartość *n* = *pq*.
- Obliczamy wartość funkcji Eulera dla *n*: $\varphi(n) = (p-1)(q-1)$.
- Wybieramy liczbę *e* (1 < *e* < *φ(n)*) względnie pierwszą z *φ(n)*.
- Znajdujemy liczbę *d*, gdzie jej różnica z odwrotnością modularną liczby e jest podzielna przez φ(*n*):

$d \equiv e^{-1} \pmod{\varphi(n)}$.
Ta liczba może być też prościej określona wzorem:
$d \cdot e \equiv 1 \pmod{\varphi(n)}$.

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

Szyfrowanie i deszyfrowanie [edytuj | edytuj kod]

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

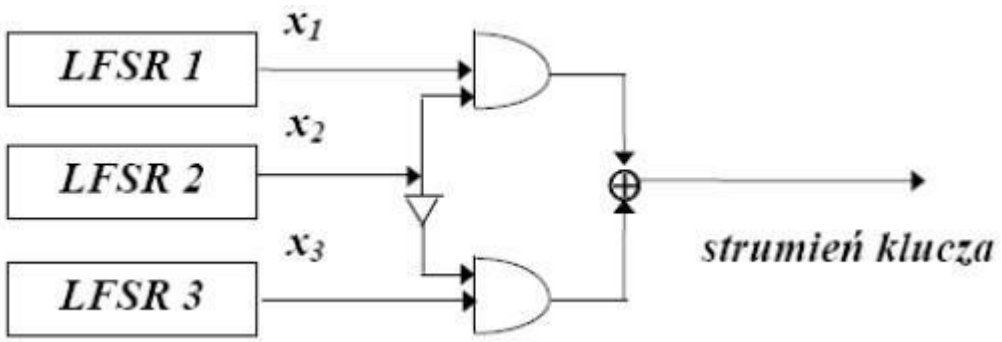
$$c \equiv m^e \pmod{n}.$$

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

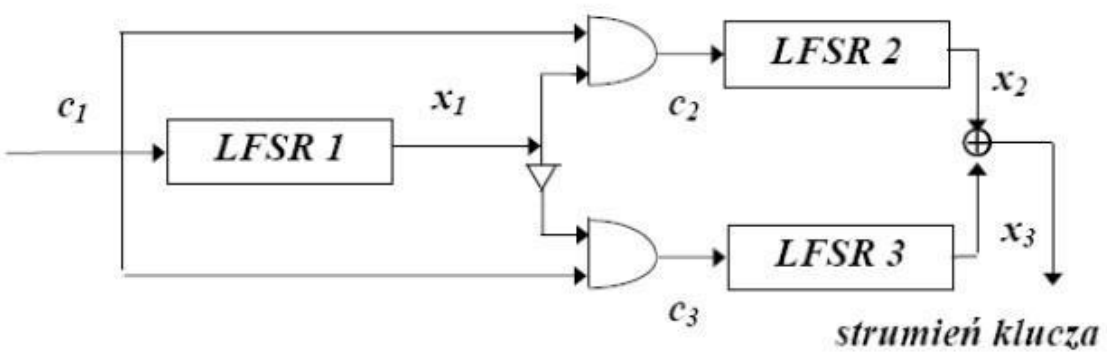
$$m \equiv c^d \pmod{n}.$$

➔ Generatory ciągów losowych:

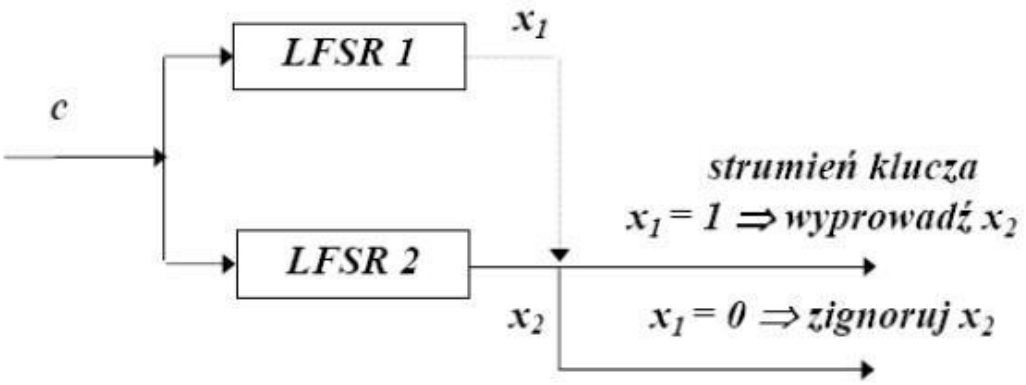
Geffego:



Stop&Go



Shrinking Generator



LFSR - Rejestr przesuwający z liniowym sprzężeniem zwrotnym (ang. linear feedback shift register, LFSR) – rejestr przesuwający, którego bit wejściowy jest funkcją liniową jego poprzedniego stanu.

➔ Demonstracja ataku AES-CBC Padding Oracle

a) Kiedy możliwy jest odczyt również pierwszego bloku?

Kiedy znany jest wektor inicjalizujący (IV). Dla szyfrowania danego bloku w CBC wykorzystuje się blok poprzedni, dla bloku pierwszego potrzebny jest więc dodatkowy ciąg. To samo dzieje się przy odszyfrowywaniu, poprzedni blok jest wykorzystywany do odczytania następnego. "Dodatkowy ciąg" to wygenerowany losowy ciąg znaków IV.

b) Jaki błąd przy implementacji należy popełnić, aby atak był możliwy?

Zwracać dodatkową informację o błędzie. Zamiast jedynie informować o tym że odszyfrowywanie się nie powiodło, zwracać informację że padding jest błędny (np. serwer zwracający kod błędu 500 jeśli padding jest niepoprawny, a 200 jeśli jest dobry).

c) W jakich środowiskach zaimplementowano ten atak? (wymień przynajmniej 3)

- ASP.NET
- SSL/TLS
- IPSEC
- SSH2

d) Czy atak działa tylko dla algorytmu AES? Odpowiedź uzasadnij.

Nie, zadziała dla algorytmów które szyfrują bazując na blokach o stałej wielkości.

e) Ile razy maksymalnie należy odpytać wyrocznię w celu odczytania jednego bloku? 255 * (wielkość jednego bloku w bajtach)

Dla AES 4080 razy.

f) Czy w przypadku zastosowania innych schematów paddingu atak będzie działał? Odpowiedź uzasadnij.

Tak, jeśli w paddingu będzie informacja o tym ile bajtów wypełnia. n - liczba wypełnionych przez padding bajtów

Przykłady innych schematów: ciąg od 1 do n, ciąg zer z ostatnim symbolem n, ostatni symbol jak n i pozostałe wybrane losowo.

➔ Metody podziału sekretu

1) Zaproponuj metodę podziału sekretu umożliwiającą rozwiązanie przedstawionego problemu i krótko opisz ją w raporcie.

Proponowana metoda to metoda wielomianu interpolacyjnego Lagrange'a, zwana też protokołem Shamir'a. Według niej pewien sekret zostaje podzielony między członków grupy i żeby go odtworzyć musi się zebrać pewna liczba członków. Wykorzystując wielomian interpolacyjny Lagrange'a wymaga się by liczba członków grupy przy której można odczytać sekret, była równa k-1 (gdzie k to liczba wszystkich członków).

Do tego zadania wykorzystano "dwustopniowy protokół Shamir'a", to jest oryginalny sekret zostaje podzielony między grupy i żeby go odczytać potrzebna jest "zgoda" wszystkich grup. Następnie część tego sekretu który trafi do grupy jest dalej rozdzielany między jej członków, a odszyfrowanie fragmentu oryginalnego sekretu jest możliwe przy zebraniu progowej liczby członków konkretnej grupy.

a) Od czego zależy bezpieczeństwo zaproponowanej metody? Bezpieczeństwo zależy od:

- Wielkości wartości progowej

Nie dotyczy to wprawdzie podanego przykładu, ale przy wartości progowej 1, nie potrzeba grupy do odczytania sekretu

- Wartości wybranych współczynników

Wybieranie tych samych współczynników w różnych grupach obniża bezpieczeństwo metody, dlatego powinny być wybierane losowo i niezależnie od siebie
- Nieparzystość

b) Czy w przypadku wykorzystania liczb typu int zaproponowana metoda będzie

bezpieczna? Odpowiedź uzasadnij.

Nie. Liczby typu int są małe, a atak brutalny wymaga sprawdzenia pm wariantów (gdzie p to wartość modulo, a m to wartość progowa). Nie jest to wprawdzie mała liczba, ale jest to realna możliwość przeprowadzenia takiego ataku z sukcesem.

➔ Badanie RSA

RSA (Rivest-Shamir-Adleman) to schemat asymetryczny algorytm szyfrowania, to jest używa klucza publicznego do szyfrowania wiadomości i klucza prywatnego do jej odszyfrowania. Klucz prywatny jest znany tylko jednej stronie, tej co odbiera wiadomość - gdyby poznał go wróg, byłoby to naruszenie bezpieczeństwa (breach of security). Klucz publiczny jest znany wszystkim. Jedną z części klucza publicznego jest wykładnik publiczny e. Jego najważniejsza cecha: jest względnie pierwszy w stosunku do modułu Ni euler(N). Nie ma jednak potrzeby generowania liczby pierwszej, można wykorzystać liczby pierwsze Fermata (3, 5, 17, 257, 65537 - jedyne znane). Użycie 65537 to dodatkowe zabezpieczenie na wypadek użycia złego padding'u dla wiadomości (problemem jest kombinacja złego padding'u (najczęściej jego braku) z małym wykładnikiem). NIST nie pozwala na użycie emniejszego niż 65537. Drugą częścią obu kluczy jest moduł N, który jest ilorazem dwóch liczb pierwszych pi q. Wartości p i qnie są przechowywane jako osobne wartości, ponieważ złamanie RSA może być tak trudne jak faktoryzacja N - to jest, czasami jest to (i dąży się do tego by było) niemożliwe. Bezpieczne RSA to niekoniecznie takie którego się nie da złamać, ale takie którego nie da się złamać w czasie nc , gdzie n = log2N , a c to mała stała (np. <5). Siła bezpieczeństwa (security strength), oznacza liczbę operacji jaką trzeba wykonać (2x) by złamać system/algorytm kryptograficzny. Podawana jest zwykle w bitach i x to wartość wybrana z następującego zestawu {80, 112, 128, 192, 256}. By zapewnić bezpieczeństwo na poziomie 256 bitów, moduł RSA musi mieć wielkość 15360 bitów. RSA rzadko kiedy nie wykorzystuje schematu podpisu, wtedy mówi się nie tylko o bezpieczeństwie samego RSA, ale bezpieczeństwie schematu podpisu. Oznacza to, że wróg nie jest w stanie wytworzyć falsyfikatu, nawet jeśli ma dostęp do podpisów wygenerowanych dla innych wiadomości. Z definicji, schemat podpisu jest niepodrabialny (bezpieczny) jeśli prawdopodobieństwo sfałszowania go przez wszystkich wrogów działających w czasie wielomianowym, jest pomijalne (istnieje pewna graniczna funkcja

nieistotności dla fałszowanego parametru, prawdopodobieństwo sfałszowania podpisu poniżej wyniku tej funkcji można pominąć). Istnieją dwa schematy podpisu RSA: PKCS1.5 i PSS. PKCS#1.5 jest deterministyczne, dla tej samej wiadomości i klucza, będzie generowany ten sam podpis za każdym razem. Jest to schemat "całkowity", przez odszyfrowanie za pomocą klucza prywatnego, można odkryć funkcję skrótu jaka została użyta do stworzenia podpisu i wyodrębnić przetworzoną wiadomość (digest). PSS jest probabilistycznym schematem podpisu; jest randomizowany przez co za każdym razem generuje inny podpis (chyba że wartość salt jest 0). W odróżnieniu od PKCS#1.5, używając PSS nie można uzyskać digest z podpisu, można jedynie porównać o znaną wartość digest.

Pomimo tego, że PSS jest bardziej złożone w implementacji, zaleca się używanie tego schematu zamiast PKCS#1.5. Dodatkowo PSS używa drugiej funkcji skrótu do wykonania padding'u: MGF (mask generation function).

Wiele czynników wpływa na to czy da się zaatakować RSA i przechwycić wiadomości czy obliczyć jeden z elementów kluczy. Na początku zostało wspomniane, że faktoryzacja modułu to problem trudny. Generowanie dużych liczb pierwszych zajmuje sporo czasu (jeśli chce się np. uzyskać bezpieczeństwo na poziomie 256 bitów). Czemu by nie wykorzystać jednego modułu dla wszystkich biorących udział w schemacie podpisu, jeśli faktoryzacja modułu jest trudna? Załóżmy, że kilku użytkowników wykorzystuje schemat podpisu do przesyłania wiadomości między sobą. Żeby uniknąć generowania nowego modułu dla każdego, można by chcieć ustalić jeden moduł N dla wszystkich użytkowników. Klucz publiczny i klucz prywatny różniłyby się wtedy tylko o wykładnik publiczny ei wykładnik prywatny d. Pomimo tego, że wydaje się że szyfrogram wyliczany dla jednej osoby X, nie mógłby być rozszyfrowany przez kogoś innego z powodu braku wykładnika prywatnego X, okazuje się to błędnym założeniem. Y (ktoś inny) może wykorzystać swoje wykładniki do faktoryzacji modułu i tym samym używając klucza publicznego X, wyliczyć wykładnik prywatny X.

➔ W zadaniu 2. zostało zadanie wykonanie kilku ataków, poniżej opisano dlaczego zadziałały.

No-message attack
Stworzenie falsyfikatu tylko za pomocą klucza publicznego, bez potrzeby uzyskania wcześniejszego podpisu. Mając klucz publiczny (e, N), generowany jest losowy podpis Si wyliczana wiadomość jako m = Semod N . Weryfikacja poprawnie wygenerowanego klucza jest zawsze poprawna, ponieważ

Se = (msgd)e = msg mod N . Dla no-message attack weryfikacja pokaże poprawny podpis, chociaż żaden podpis dla wiadomości nie został wygenerowany. Przy wykorzystaniu funkcji skrótu H na wiadomości problem staje się złożony: jeśli msghash = Se mod N , to trzeba znaleźć taką wiadomość msg ∈ {0, 1}* żeby H(msg) = msghash . Jeśli funkcja nie jest wydajnie odwracalna jest to problem trudny.

Fałszowanie podpisu na wiadomościach
Wybrane są dwie wiadomości msg1 i msg2takie że msg = msg1 msg2. Wróg generuje podpisy S1 i S2. Dzięki poniższej równości można stwierdzić, że S = S1S2 mod N jest prawidłowym podpisem dla msg.

Se = (S1S2)e = (msgd1msgd2)e = msg1edmsg2ed = msg1msg2 = msg mod N

Jest to fałszerstwo, ponieważ żadna ze składowych wiadomości nie jest równa tej właściwej. Przy użyciu funkcji skrótu H trzeba rozwiązać poniższą równość:

H(msg) = H(msg1) × H(msg2) mod N

Tak samo jak w przypadku no-message attack, jeśli funkcja nie jest wydajnie odwracalna jest to problem trudny.

➔ Hastad's Broadcast Attack

Ten atak wykorzystuje w praktyce twierdzenie Coppersmit'a. Jedna wiadomość *msg*jest szyfrowana trzy razy za pomocą trzech klucz publicznych - wykładnik publiczny jest wspólny *e*= 3, a klucz publiczny jest dla każdego inny. Dodatkowo klucze publiczne muszą być między sobą parami względnie pierwsze. Wyliczane są trzy szyfrogramy z wzoru książkowego dla RSA: $C_i = msg^3 \ mod \ N_i$. Można zapomnieć wiadomość. Wykorzystując chińskie twierdzenie o resztach, można znaleźć taki szyfrogram, że: $cipher = msg^3 \ mod \ N_1 N_2 N_3$. Jeśli *msg*jest mniejsze od iloczynu wszystkich modułów, to $msg^3 < N_1 N_2 N_3$, co znaczy że wyliczając pierwiastek sześcienny z *cipher*można otrzymać oryginalną wiadomość.

➔ Atak na szyfrowanie RSA

Moduł ze względu na swój rozmiar uniemożliwia rozłożenie go na czynniki pierwsze, obliczenie więc p i q, a tym samym d jest niemożliwe. Mały wykładnik publiczny pozwala jednak na wyliczenie wiadomości z szyfru. Okazuje się, że

można wyliczyć $C = M^e \ mod \ N$ jako $\sqrt[3]{C}$ (jeden z analizowanych problemów

RSA, jak wyliczyć pierwiastek e-tego stopnia z $C \ mod \ N = pq$, jeśli faktoryzacja *N*nie jest znana). Obliczenie pierwiastka modulo m jest równoważne rozłożeniu na czynniki liczby m - problem obliczeniowo trudny. Zamiast tego, łatwiej jest znaleźć liczbę która podniesiona do trzeciej potęgi da C. Dotyczy ataku na szyfrowanie i Hastad's broadcast attack. Używając funkcji skrótu H i pracując na tym skrócie, można przeprowadzić atak, ale na samym końcu rozwiązać równość: $H(msg) = msghash$.

Jeśli funkcja nie jest wydajnie odwracalna jest to problem trudny.

Generowanie strumieni klucza przy użyciu szyfratorów strumieniowych

Testy dla ciągów binarnych:

a. Monobit test
   Żeby test przeszedł pomyślnie, liczba jedynek powinna mieścić się w zakresie (9725; 10275).
a. Poker test
   Ciąg binarny dzielony jest na 5000 bloków po 4 bity i liczone jest wystąpienie każdej z możliwych (binarnych) kombinacji. Obliczana jest wartość funkcji:

   $(16/5000) \times (\sum_{i=0}^{15} [f(i)]^2) - 5000$. Jeśli jej wartość mieści się między (2.16; 46.17), test przeszedł pomyślnie.
a. Long run test
   Żeby test przeszedł pomyślnie, nie powinno być w ciągu zer lub jedynek 26 razy pod rząd.

➔ Przykłady innych ataków RSA to:

Partial key exposure attack

Jeśli wykładnik publiczny $e < \sqrt{N}$ i znana jest części najmniej znaczących bitów wykładnika prywatnego d pozwala na odszyfrowanie całego d.

a.

Bleichenbacher's attack on PKCS1

Atak podobny do *padding oracle attack*.Załóżmy że moduł jest o rozmiarze n-bitów, a wiadomość jest o rozmiarze m-bitów i m < n. Do wiadomości M jest dodany padding by dopełnić go do rozmiaru n-bitów i całość jest szyfrowana. Schemat PKCS1 na samym początku dostawia blok "02" (długość 16 bitów), który informuje o tym że losowy padding został dodany przed wiadomością. Niektóre aplikacje będą miały *padding oracle*, który zwróci wiadomość o braku bloku "02". Tak jak w *padding oracle attack*można testować wielkość paddingu do skutku i odszyfrować szyfrogram.

Coppersmith's short pad attack

Do wiadomości M jest dodawany padding, kilka losowych bitów na końcu. Wysłana jest jedna wiadomość, która zostaje przechwycona przez wroga. Naiwnie druga wiadomość o tej samej treści, z padding'iem dodany w ten sam sposób zostaje wysłana. Wróg przechwycając drugą wiadomość, ma teraz dwie te same wiadomości z padding'iem różnym o kilka bitów. Nie tylko wróg może odczytać szyfrogram, ale też początkową wiadomość.

Pokazane przykłady pokazują jak można wykorzystać i atakować słabości źle zaimplementowanego schematu podpisu RSA. Nie zawsze trzeba znać wszystkie składowe kluczy (zwłaszcza prywatnego) czy rozłożyć moduł, by skompromitować schemat - czasem wystarczy podstęp. Jednak RSA nie byłby tak powszechnie stosowany, gdyby dało się go tak łatwo złamać. Bezpieczny schemat podpisu RSA to taki, który implementuje kilka poziomów bezpieczeństwa i w przypadku przebicia się wroga przez jeden, nadal nie pozwala na wyciek wrażliwych danych. Nie trzeba ukrywać wszystkich elementów schematu, ale trzeba chronić nawet te które nie są wszystkim znane.

➔ Jakie właściwości powinny posiadać schematy grupowych podpisów cyfrowych?

➔ Jakie zbiory wielomianów są wykorzystane jako podstawa matematyczna systemów kryptograficznych NTRU (N-truncated polynominals) ? Proszę wyjaśnić „logikę" dwóch podstawowych operacji binarnych dla tych wielomianów.