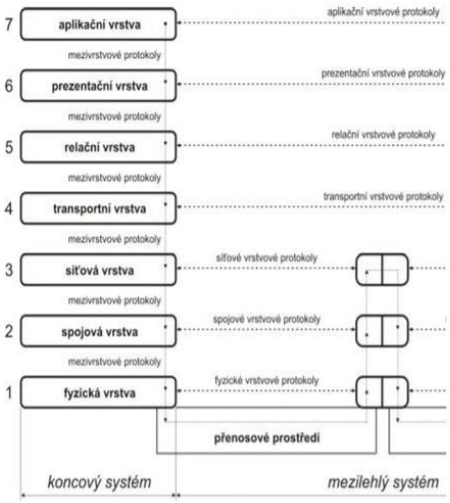
1. *Popište, čím se zabývá tzv. „Architektura bezpečnosti v RM OSI“, které částí obsahuje a jakým způsobem se implementuje.*

- *Jedná se o doporučení ITU-T X.800, ISO 7498-2 ISO/OSI Secure Architecture*
- *Obsahuje:*
    - *Služby bezpečnosti (security services) – definované postupy pro zabezpečení informačních systémů*
    - *Mechanismy bezpečnosti (security mechanism)*
    - *Útoky na bezpečnost (security attacks)*
- *Bývají zabudované do aplikačních protokolů (aplikačních programů) a transportů dat (operačních systémů) - 7. a 4. vrstva - nebo do propojovacích zařízení - 3.vrstva – síťová vrstva – směrování*

Vrstvová architektura ISO RM OSI



7. zajišťuje celou řadu služeb, např. přenos souborů, el. pošta, vzdálené zadávání úkolů, … **rozhraní na aplikační prostředí.**
6. provádí často používané aplikační funkce související se syntaxí přenášených dat, např. kódování, **šifrování**, komprese, … **sjednocení kódování.**
5. správa dialogu, předávání pověření, synchronizace mezi koncovými uživateli (procesy), **spolupráce mezi procesy.**
4. řešení komunikace koncových uživatelů, řízení toku, řízení síťového spojení (TCP) **adresovatelné rozhraní.**
3. **směrování toku dat sítí** (uspořádaných do paketů) komunikační cesta mezi adresovatelnými nesousedními partnery.
2. zajišťuje vytvoření **spolehlivého bezchybného datového spoje mezi sousedními uzly**, řízení toku rámců, přístup k médiu
1. **zajišťuje vysílání a příjem signálů nesoucí informaci**, přenos bitů, zaobírá se mech. el. a procedurálním rozhraním přenosového média.

11

- *Služby bezpečnosti – ISO 7498-2 – 5 kategorií služeb*
    - *Autentizace – authentication*
        - *Uživatelů – neeliminují útoky zopakováním zpráv*
        - *Zdroje dat – provádí autentizaci všech dat – eliminují útoky zopakování zpráv*
    - *Řízení přístupu – access control*
        - *Přístup do systému – ochrana před neautorizovaným přístupem (operační systém nebo aplikační program)*
    - *Zabezpečení důvěrnosti dat – data confidentiality*
        - *Služby pro důvěrnost přenosu zpráv*
        - *Služby pro důvěrnost spojení – ochrana v rámci navázaného spojení*
        - *Služby pro důvěrnost toku dat – chrání informace na základě atributů toku dat*
        - *Služby selektivní důvěrnosti – ochrana pouze určených částí informace*
    - *Zabezpečení integrity dat – data integrity*
        - *Zabezpečení proti neautorizované modifikaci*
            - *Služby integrity přenosu zpráv – ochrana int. přenášených zpráv*
            - *Služba integrity spojení – ochrana přenosů v rámci určitého navázaného spojení*

        - *Služby selektivní integrity spojení a selektivní integrity zpráv*
    - *„Slabá“ integrita – pro objektivní útoky*
        - *Modifikace zprávy šumem, náhodná změna pořadí paketů, náhodná duplicita – aplikace kontrolních součtů, CRC, pořadová čísla paketů*
    - *„Silná“ integrita – subjektivní (úmyslné, aktivní útoky)*
        - *Podvržené zprávy, úmyslné pozměněné zprávy – prostředky pro zajištění slabé integrity + kryptografické prostředky*
            - *Služby integrity bez oprav - detekce porušení integrity*
            - *Služba integrity s opravami – obnova integrity po detekci ztráty integrity*
- *Ochrana proti odmítnutí původu zprávy - non-repudation*
    - *Zajišťuje důkaz o původu dat*
    - *Prokázání původu – příjemce a odesílatel*
    - *Prokázání doručení – odeslání a přijetí*
- *Autentizace a nepopiratelnost*
    - *Autentizace – vím s kým komunikuji*
    - *Nepopiratelnost – vím s kým komunikuji a lze mu to dokázat*

2. *Uveďte a stručně charakterizujte:*
    - *služby bezpečnosti zajišťované kryptografickými prostředky,*
        - *Autentizace*
        - *Řízení přístupu*
        - *Zabezpečení důvěrnosti dat*
        - *Zabezpečení integrity dat*
        - *Ochrana proti odmítnutí původu zprávy*

    - *kryptografické mechanismy, které tyto služby zajišťují.*
        - *Šifrování – encipherment*
        - *Digitální podpis – digital signature*
        - *Řízení přístupu – access control*
        - *Integrita dat – data integrity*
        - *Výměna autentizační informace – authentication exchange*
        - *„Výplň“ – traffic padding*
        - *Řízení směrování – routing control*
        - *Ověření třetím subjektem – notarization*

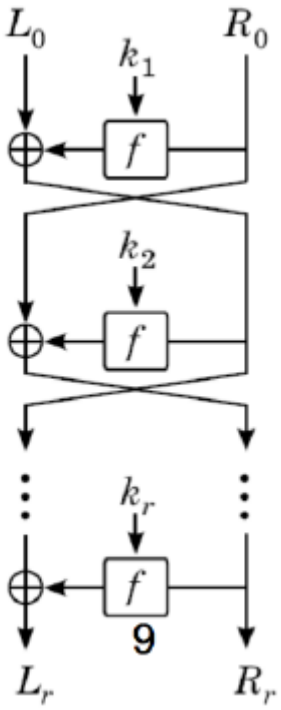
3. *Uveďte jednotlivé kroky pro zajištění bezpečné komunikace.*
    - *Navázání spojení s **autentizací** prostřednictvím asymetrického kryptografického algoritmu*
    - *Výměna symetrických klíčů pro zajištění **integrity a důvěrnosti** následné výměny zpráv*
    - ***Bezpečná výměna zpráv***
    - *Zrušení spojení včetně všech zbytkových informací*
    - ***Ověření autentičnosti, integrity a důvěrnosti** přijaté informace*

4. *Vysvětlete význam pojmů souvisejících s modely hrozeb: Destruction, Corruption, Removal, Disclosure, Interruption.*
    - *Destruction (útok na dostupnost) – zničení dat či síťových zdrojů*
    - *Corruption (útok na integritu) – neautorizovaná modifikace aktiv/dat*
    - *Removal (útok na dostupnost) – krádež, odebrání či ztráta informací nebo jiných zdrojů*
    - *Disclosure (útok na důvěrnost) – neautorizovaný přístup k aktivům/datům*
    - *Interruption (útok na dostupnost) – přerušení služeb, spojení začne být nepoužitelné*

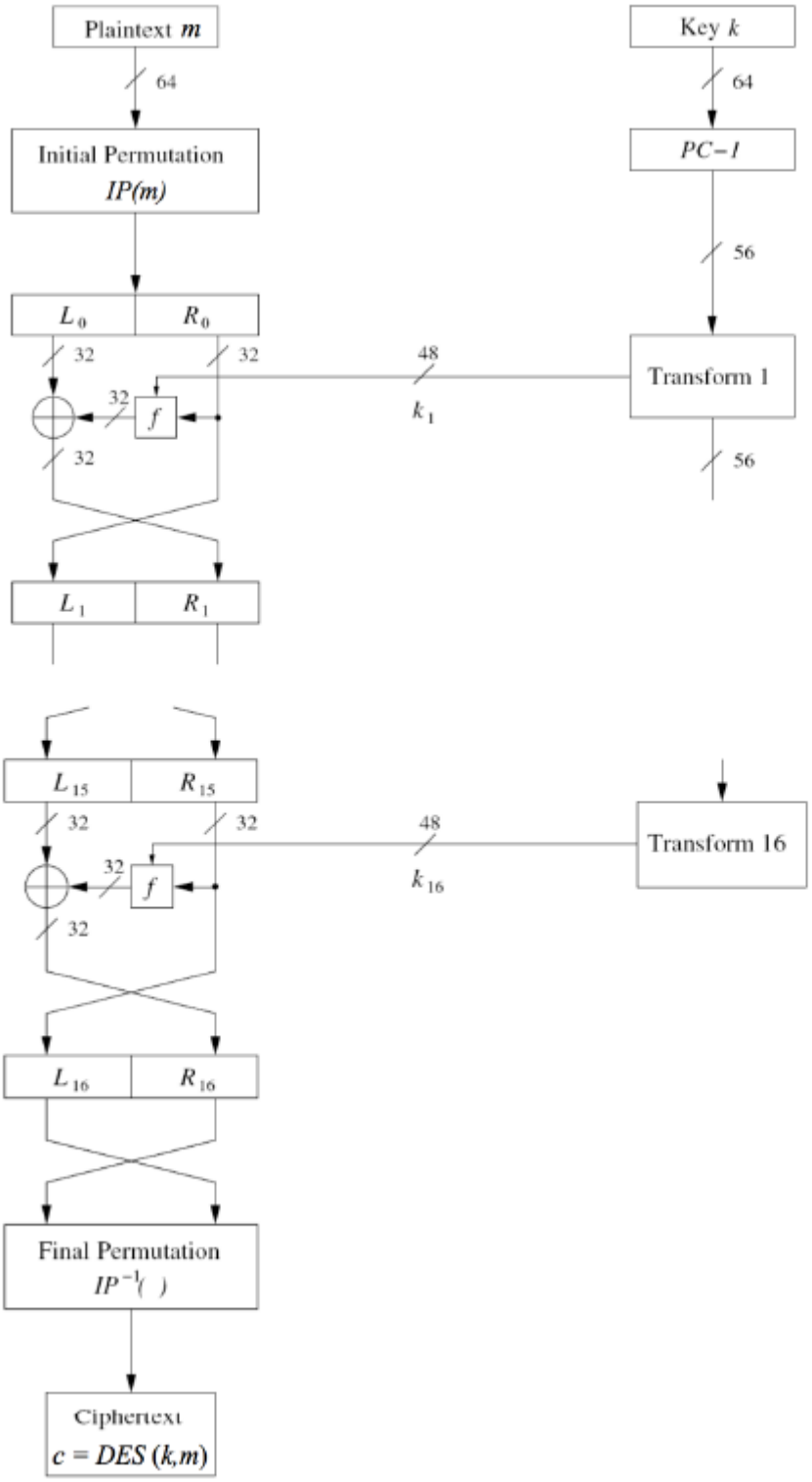
5. *Synchronní a asynchronní proudové šifry, základní charakteristiky.*
    - *Proudové šifry*
        - *Šifrování bit-bit, případně byte-byt, generátor vytváří tajné heslo*
        - *Výhody*
            - *Rychlost*
            - *Malé hlášení chyb*
        - *Nevýhody*
            - *Nízká úroveň difúze*
            - *Nebezpečí použití stejného hesla dvakrát*
        - *Používané na zařízeních s malou pamětí*
    - *Synchronní proudové šifry*
        - *Proud hesla je generovaný nezávisle na vstupu(M)a šifrovém textu(C)*
        - *Následně dojde ke kombinaci vygenerovaného hesla s M a C, nejčastěji pomocí kombinace keystreamu a textu je použití XOR operace (binární doplňková proudová šifra)*
        - *Odesílatel i přijímatel musí být v jednotlivých krocích dešifrování přesní, aby vše proběhlo úspěšně (musí být synchronizovaní)*
        - *Pokud jsou čísla během přenosu přidané nebo odebrané synchronizace je ztracená*
        - *Pokud se však ztratí pouze jedna z číslic je tato část ovlivněna, ale zbytek zprávy to neovlivní*
    - *Asynchronní (samosynchronní) proudové šifry*
        - *Šifrovací klíč je závislý na šifrovaném textu – vygenerován z n čísel z C*
        - *Při ztrátě synchronizace dojde po určité době k opětovné synchronizaci*

6. *Co se rozumí pod pojmem "Product Ciphers" objasněte princip konstrukce.*
    - *Kaskádová šifra*
        - *Konstrukce většiny dnešních blokových šifer*
        - *Opětovně se střídá konfúze a difúze*

7. *Co je to „Feistelova síť", znázorněte graficky, stručně charakterizujte.*
    - *Princip symetrické blokové šifry*
    - *Vstupní blok o velikosti n bitů rozdělíme na 2 části – L a R*
    - *Interaktivní algoritmus, kde vstup do i-té rundy je výstupem předcházející rundy*
    - *Reverzibilnost funkce je dána funkcí XOR, bez ohledu na složitost funkce*



8. *Základní charakteristiky šifer DES, 3DES, AES.*
    - *DES – Data encryption standard*
        - *Bloky o velikosti 64 bitů, klíč – 56 bitů, 16 rund*
        - *Zvláštní klíč pro každou rundu*
        - *V dnešní době rozluštitelný hrubou silou (brute force) –> DES-cracker*
        - *Zesílení triple DES*
        - *Používá Feistelovu síť*
            - *Blok zprávy 64 bitů je rozdělen do dvou registrů po 32 bitech*
            - *Výstup 32b R_i je přiveden do funkce které výstup 32b je XOR 32b L_i*
            - *Následuje přehození hodnot pravého registruR_{i-1} s levým L_i*
            - *Postup se opakuje až do 16 rundy, po které následuje konečná permutace (inverzní k počáteční)*
            - *Šifrování a dešifrování se liší jen v pořadí použití rundového klíče*



- *3DES*
  - *Umělé zesílení DES*
  - *Prodloužení klíče na 56 + 56 (+ 56) bitů*
  - *Používá se všude tam kde je potřeba schválený bezpečný algoritmus a nevadí zpomalení*
  - *3DES_112, 3DES_168*

- *AES – Advanced Encryption Standard*
  - *Počet rund záleží na délce klíče*
  - *Bajtově orientovaná šifra*
  - *Šifruje i dešifruje data stejným klíčem, které jsou rozděleny do bloků pevně dané délky 16B*
  - *Blok je matice s rozměrem 4x4B – velikost klíčů se pohybuje od 16B do 64B*
  - *Používá se pro bezdrátové Wi-Fi sítě v rámci zabezpečení WPA2*
  - *Postup algoritmu:*
    - *Expanze klíče (Key Expansions)*
      - *Podklíče jsou odvozeny z klíče šifry užitím Rijndael programu*
    - *Inicializační část (Initial Round)*
      - *Každý byte stavu je zkombinován s podklíčem za pomocí operace XOR nad všemi bity*
    - *Iterační část (Rounds)*
      - *Záměna bitů (SubBytes) – nelineární nahrazovací krok, kde je každý byte nahrazen jiným podle vyhledávací tabulky (S-box)*
      - *Prohození řádku (Shift Rows) – provedení kroku, ve kterém je každý řádek stavu postupně posunut o určitý počet kroků*
      - *Kombinování sloupců (Mix Columns) – zkombinuje čtyři byty v každém sloupci*
      - *Přidání podklíče (Add Round Key)*
    - *Závěrečná část*
      - *Záměna bytů (SubBytes)*
      - *Prohození řádků (Shift Rows)*
      - *Přidání podklíče (Add Round Key)*

9. *Znázorněte a stručně charakterizujte módy blokových šifer: CBC, CFB, OFB, CTR, MAC.*
   - *ECB – Electronic Code Book*
     - *Každý blok je šifrován samostatně*
     - *Opakovaný blok je shodně šifrován*
     - *Pro krátké zprávy, rozesílání klíčů*
     - *Vhodné pro poruchové spoje*
     - *Nezajišťuje integritu otevřeného textu*
   - *CBC – Cipher Block Chaining*
     - *Řetězení šifrového textu – rozšíření difúze a konfúze*
     - *Každý blok je před šifrováním XOR s předchozím zašifrovaným blokem*
     - *První blok XOR s IV inicializačním blokem, tzv nonce („number used once“), který se přijímací straně vyšle otevřeně*
     - *Poslední blok doplněn na potřebnou délku, kontrola při dešifrování*
     - *Po ztrátě bloku šifrového textu se synchronizuje po přijetí dvou bloků*
   - *CFB – Cipheretext FeedBack*
     - *Využití blokové šifry jako zdroje „hesla“ pro proudovou šifru*
     - *Zdroj (generátor) hesla ovlivněn zpětnou vazbou branou ze zašifrovaného textu*
     - *Schopnost „samosynchronizace“*
     - *Při dešifrování náchylné na chybovost spoje*
   - *OFB – Output FeedBack*
     - *Zpětná vazba zavedená z výstupu samotného „generátoru“*
     - *Vlastnosti synchronní proudové šifry*

- *Pro poruchové spoje, satelitní komunikace*
- *CTR – Counter mode*
  - *Obdoba OFB*
  - *Převádí blokovou šifru na synchronní proudovou šifru*
  - *Heslo lze vypočítat na základě pozice otevřeného textu*
  - *Nemá vlastnost samosynchonizace*
- *MAC*
  - *Autentizační kód zprávy*
  - *Zajištění integrity, obdoba CBC*

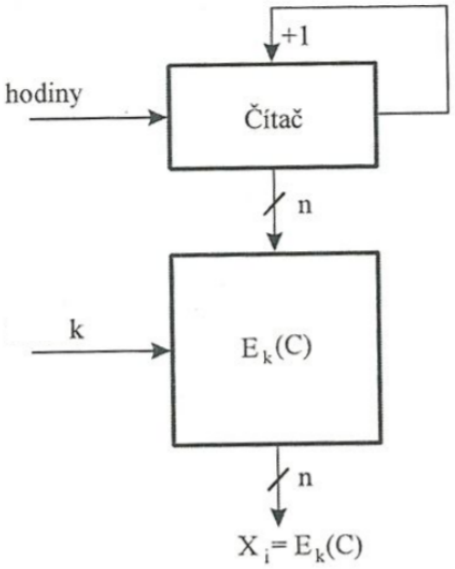
10. *Jak souvisí hodnota entropie s bezpečností kryptografických generátorů náhodných čísel.*
    - *Čím větší je hodnota entropie (neurčitosti), tím méně (z definice) víme nebo můžeme odhadovat, jaké (pseudo)náhodné číslo generátor vytvoří a tím je generátor bezpečnější*
    - *Čím větší je entropie, tím více si jsme jistí, ze pseudonáhodné číslo nemůže nikdo odhadnout a tím prolomit zabezpečení*

11. *Jaké jsou požadavky kladené na kryptograficky bezpečné generátory náhodných čísel.*
    - *Standartní rovnoměrné rozložení*
    - *Statistická nezávislost*
    - *Nepředvídatelnost*
    - *Rychlé generování*

12. *Znázorněte princip generování náhodných čísel s využitím:*
    - *Symetrické šifry*

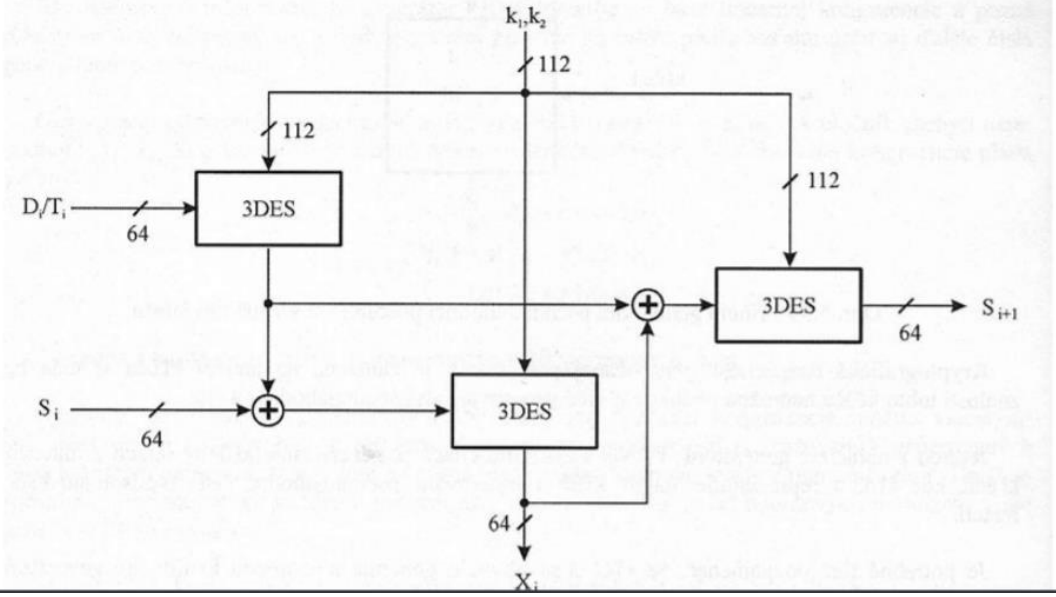
klíč *k* - náhodná data (*seed*)

generovaná posloupnost $X_0, X_1, X_2, \ldots$



Generátor podle ANSI X 9.17

$$X_i = EDE_{k_1,k_2}\left[S_i \oplus EDE_{k_1,k_2}\left(D_i / T_i\right)\right]$$
$$S_{i+1} = EDE_{k_1,k_2}\left[X_i \oplus EDE_{k_1,k_2}\left(D_i / T_i\right)\right]$$



- *Hašovací funkce*
  - *Např. PKCS#1 v.2.1 definuje pseudonáhodný generátor MGF1 (Mask Generator Functionú pomocí hashovací funkce H s pořátečním – většinou náhodným – nastavením seed takto:*
    - *H(seed || 0x00000000), H(seed || 0x00000001), H(seed || 0x00000002), H(seed || 0x00000003),….*
  - *Tvorba klíče DK z passwordu podle PKCS#5*

PBKDF1 (P, S, c, dkLen)

Hash *underlying hash function*

P *password, an octet string*

S *salt, an eight-octet string*

c *iteration count, a positive integer*

dkLen *intended length in octets of derived key, at most 16 for MD2 or MD5 and 20 for SHA-1*

DK *derived key, a dkLen-octet string*

$T_1 = Hash\ (P\ ||\ S)$,

$T_2 = Hash\ (T_1)$,

…

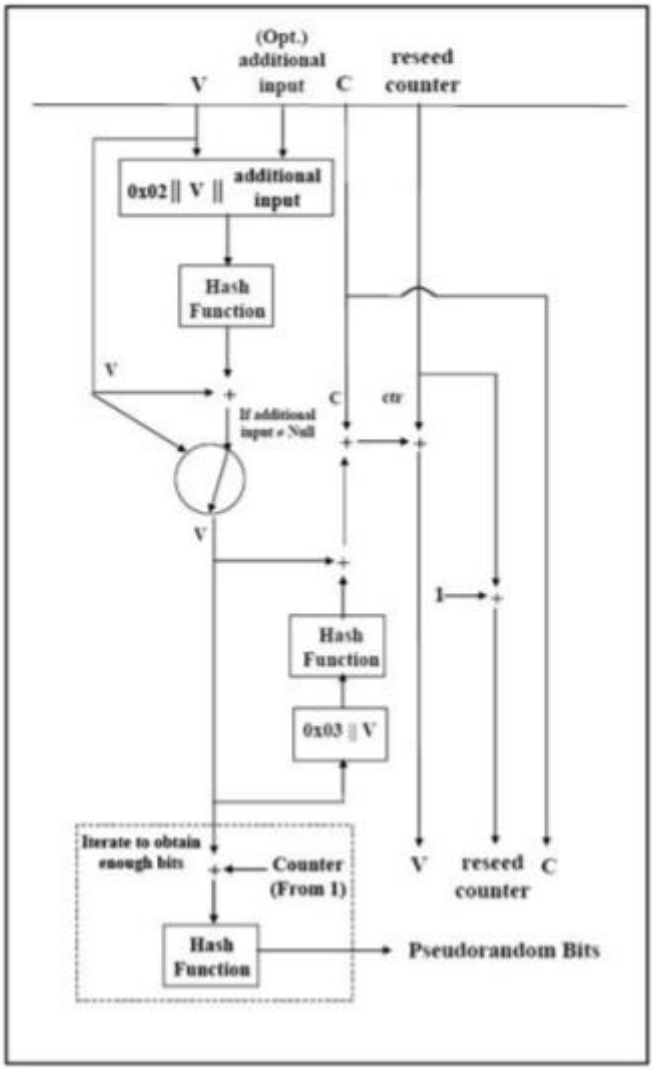
$T_c = Hash\ (T_{c-1})$,

$DK = T_c\langle 0..dkLen\text{-}1\rangle$

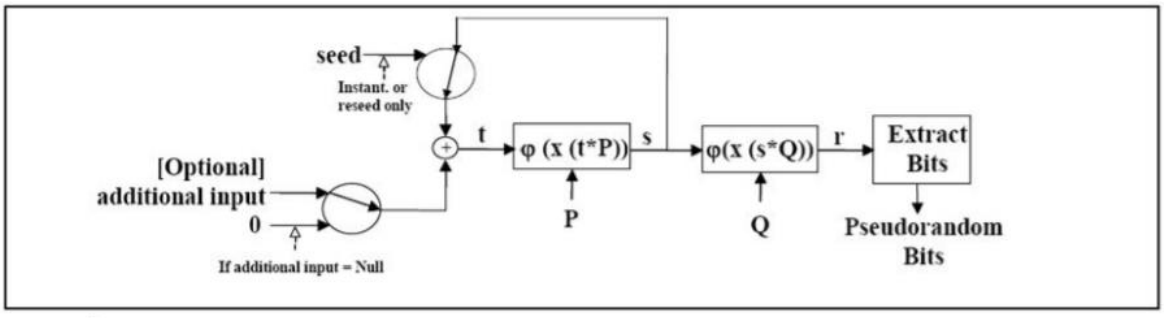
Hash Function	Block Cipher	OID
MD2	DES	pkcs-5.1
MD2	RC2	pkcs-5.4
MD5	DES	pkcs-5.3
MD5	RC2	pkcs-5.6
SHA-1	DES	pkcs-5.7
SHA-1	RC2	pkcs-5.8

13. *Uveďte příklady realizace kryptograficky bezpečných generátorů náhodných čísel.*

- *PRGN s hašovací funkcí dle NIST 800-90A*



- *PRGN založený na ECC dle doporučení NIST 800-90A*



14. *K čemu se využívá „kvantový přenos informace“. Uveďte důvody použití, protokol BB84.*
    - *Nemožnost vytvoření identické kopie na základě neznámého kvantového stavu (vycházíme z Heisenbergova principu neurčitost)*
    - *Čtení zprávy zároveň ovlivňuje její obsah*
    - *Kvantový bit = quibit – může nabývat nekonečně mnoho hodnot mezi hodnotami 0 a 1, avšak měřením jednoho quibitu získáme nejvýše jeden bit klasické informace*
    - *Využití:*
        - *Bezpečná komunikace – nepodmíněná bezpečnost*
            - *V kombinaci se symetrickou šifrou (Vernamova šifra)*
            - *Výměna klíčů – náhrada asymetrických systémů*
        - *Generování náhodných čísel*
    - *Protokol BB84*
        - *Slouží k dohodě na symetrickém klíči (následně použit např. pro systém jednorázového hesla)*
        - *Neřeší autentizaci uživatelů*
        - *Foton nejde rozdělit*
        - *Nelze vytvořit přesnou kopii*
        - *Eva se tváří jako Bob a Alice, nezná polarizační bázi, způsobí v přenosu průměrně 50% chyb*
        - *Stálý odposlech způsobí v přenosu průměrně 25% chyb*

I. Kvantový přenos

1. Alice vybere náhodné bity.
2. Alice náhodně vybere vysílací polarizační báze.
3. Alice kóduje bity do polarizací posílaných fotonu.
4. Bob náhodně vybírá přijímací polarizační báze.
5. Bob zaznamenává obdržené bity
   (některé fotony se ztratí - nejsou detekovány).

II. Veřejná diskuse

6. Bob oznamuje báze, ve kterých naměřil fotony.
7. Alice oznamuje, které báze byly správně „uhodnuty“.
8. Shodli-li se Alice a Bob v bázích, přenesený bit si ponechají.
   (nenaslouchala-li Eva má Bob přesně to, co Alice poslala).

III. Obětování bitu

9. Bob obětuje některé náhodně vybrané bity k odhalení Evy.
10. Alice potvrzuje tyto obětované bity (Eva by způsobila odchylky).
11. Zbylé tajné bity sdílené Alicí a Bobem tvoří klíč.

15. *Vysvětlete, co je digitální podpis, jaké jsou na něho kladené požadavky a jakým způsobem se používá v součinnosti s hašovací funkcí.*
    - *Digitální podpis = podpis vytvořený kryptografickými prostředky*
        - *Měl by být nefalšovatelný*
        - *Prostředek autentizace (jednoznačně přiřazen uživateli)*
        - *Nepřenosný*
        - *Podepsaný dokument není možné měnit*
        - *Podpis nelze popřít*
    - *Použití v součinnosti s hashovací funkcí – nepodepisujeme celý dokument, ale pouze hash -> v případě úpravy dokumentu se výrazně změní i hash*

16. *Vysvětlete, co je časové razítko, jakým způsobem se vytváří, jaké jsou na něho kladené požadavky.*
    - *Časové razítko = struktura obdobná certifikátu, která svažuje kontrolní součet (hash) z dokumentu s časem*
    - *Časové razítko je elektronicky podepsáno (vydáváno) autoritou pro vydávání časových razítek (TSA)*
    - *Elektronicky podepsaná struktura časového razítka obsahuje:*
        - *Jméno vydavatele (jméno TSA)*
        - *Jedinečnost sériového čísla razítka*
        - *Kontrolní součet (hash) z dokumentu a čas*
        - *Důvěryhodný zdroj času – ideálně 3 nezávislé zdroje času*

17. *Co se rozumí pod pojmem "Public Key Infrastructure", jakým způsobem lze zajistit.*
    - *Infrastruktura veřejných klíčů*
    - *Souhrn technických a organizačních prostředků spojených s vydáváním, správou, používáním a odvoláváním platnosti kryptografických klíčů a certifikátů*
    - *Zabraňuje použití falešné identity*
    - *Veřejný klíč je platný pouze v případě potvrzení důvěryhodnou stranou, například certifikační autoritou*

18. *Vysvětlete funkci "Public Key Infrastructure" na základě certifikátů.*
    - *Veřejný klíč je platný pouze v případě potvrzení důvěryhodnou stranou, například certifikační autoritou*
        - *Třída 1*
            - *Ručí pouze za jednoznačnost certifikátu*
            - *Žadatel vyplní formulář serveru a protokolem HTTPS ji odešle*
        - *Třída 2*
            - *Ručí za jednoznačnost certifikátu + kontroluje totožnost uživatele*
            - *Vybudovaná síť registračních autorit, kam osobně chodí uživatelé se svými žádostmi*
            - *RA ověří totožnost uživatele a odešle žádost k CA*
            - *CA uchovává svůj soukromý klíč v bezpečném HW*
        - *Třída 3*
            - *Stejné jako třída 2, ale vydané certifikáty jsou určené výhradně pro konkrétní aplikaci*
            - *Např certifikát vydaný bankou lze použít na přihlašování do internetového bankovnictví, ale není možné ho použít na podepisování emailu*
            - *CA uchovává svůj soukromý klíč v bezpečném HW*

19. *Certifikát podle X.509, k čemu slouží, co obsahuje.*
    - ***Verze**: upřesňuje formát certifikátu*

- ***Sériové číslo**: jednoznačně identifikují číslo přidělené od CA, každý certifikát vydaný CA má jiné sériové číslo*
- ***Identifikátor algoritmu:** algoritmus, spolu s dalšími údaji použitými při vytváření podpisu certifikátu*
- ***CA:** identifikátor CA, která vytvořila a podepsala certifikát*
- ***Doba platnosti:** obsahuje dva časové údaje – časový okamžik, před kterým certifikát ještě neplatí a časový okamžik, po kterém už neplatí*
- ***Subjekt:** uživatel, kterému patří certifikovaný veřejný klíč*
- ***Veřejný klíč:** podepisovaný veřejný klíč, spolu s identifikátory algoritmu, pro které je určen*
- ***Podpis CA:** je funkcí všech položek certifikátu*
- *Je určen pro jednoduché podepisování*

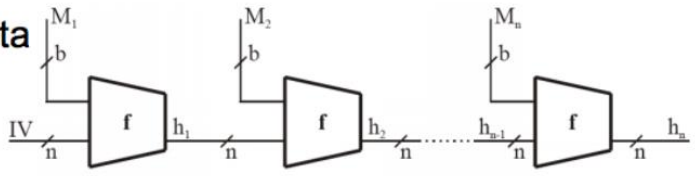
20. *V souvislosti s nařízením eIDAS vysvětlete pojmy:*
    - *Elektronický podpis*
        - *Data v elektronické podobě, která jsou připojena k jiným datům v elektronické podobě nebo jsou s nimi logicky spojena a která podepisující osoba používá k podepsání*
    - *Zaručený elektronický podpis*
        - *Elektronický podpis, který splňuje požadavky stanovené v článku 26*
            - *Jednoznačně spojen s podepisující osobou*
            - *Umožňuje identifikaci podepisující osoby*
            - *Je vytvořen pomocí dat pro vytváření elektronických podpisů, která podepisující osoba může s vysokou úrovní důvěry použít pod svou výhradní kontrolou*
            - *Je k datům, která jsou tímto podpisem podepsána, připojen takovým způsobem, že je možné zjistit jakoukoliv následnou změnu dat*
    - *Kvalifikovaný elektronický podpis*
        - *Zaručený elektronický podpis, který je vytvořen kvalifikovaným prostředkem pro vytváření elektronických podpisů a který je založen na kvalifikovaném certifikátu pro elektronické podpisy*
    - *Elektronická pečeť*
        - *Data v elektronické podobě, která jsou připojena k jiným datům v elektronické podobě nebo jsou s nimi logicky spojena s cílem zaručit jejich původ a integritu*
    - *Elektronické časové razítko*
        - *Data v elektronické podobě, která spojují jiná data v elektronické podobě s určitým okamžikem a prokazují, že tato jiná data existovala v daném okamžiku*

21. *Vysvětlete princip konstrukce iteračních hašovacích funkcí, jaké jsou na ně kladené požadavky, jak se hodnotí jejich bezpečnost, příklady algoritmů.*
    - *Využívají speciální kompresní funkci f, která je funkcí dvou proměnných (dva vstupy):*
        - *n-bitový vstup z předcházejícího kroku*
        - *b-bitový blok vstupní zprávy*
        - *MD5, SHA1, SHA2*

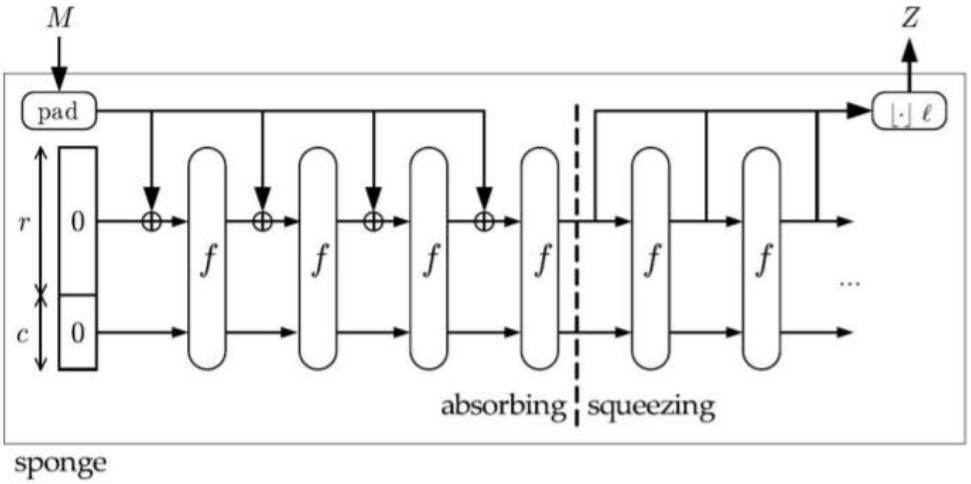
$IV = h_0$ = počáteční hodnota

$h_i = f(h_{i-1}, m_i)$

$h = h_n$



22. *Vysvětlete princip konstrukce hašovacích funkcí typu „houba“ (SHA3), příklady využití.*
    - *Zpracovává zprávu M po blocích*
    - *f – permutace (kompresní funkce)*
    - *c – tzv. kapacita, r – bitová rychlost, b=r+c*
    - *Konstrukce houby tvoří kryptografické primitivum využitelné pro různé aplikace*



SHA-3 - Keccak

- 7 variant permutací: b {25, 50, 100, 200, 400, 800, **1600**}
- varianty SHA3 – n, n - výstup
- SHA3-224: r = 1152; c = 448;
- SHA3-256: r = 1088; c = 512;
- SHA3-384: r = 832; c = 768;
- SHA3-512: r = 576; c = 1024;
- SHA 3: b = 1600, r = 1088 and c = 512

- odlehčená verze: b = 200, r = 40, c = 160

23. *Vysvětlete princip jednorázového podpisu pomocí hašovacích funkcí (Lamport), uveďte jeho výhody a nevýhody.*
    - *Podepisování každého bitu zvlášť*

```
# JEDNOBITOVÁ ZPRÁVA
vygenerujeme k_prv: 256b řetězce S0, S1
spočítáme    k_pub: H0 = h(S0), H1 = h(S1)
zveřejníme H0, H1
pro podepsání b'0' zveřejníme s=S0
ověření: h(s) = H0

# N-BITOVÁ ZPRÁVA
k_prv je dvojice řetězců [S0-1, S0-2, S0-3, ...; S1-1, S1-2, S1-3, ...]
k_pub je dvojice řetězců [H0-1, H0-2, H0-3, ...; H1-1, H1-2, H1-3, ...]
každý bit se podepíše odděleně
veřejný i soukromý klíč mají velikost 2n² bitů
```

    - *Merkleho strom pro opakované podpisy*
        - *Aby nebyly veřejné klíče obrovské zahashujeme je Merkleho stromem a zveřejníme jeho vrchol*
        - *S podpisem zveřejníme jen ty, které jsou potřebné*
        - *Alice pošle zprávu L2 a doplňující hashe 0-0 a 1. Bob Top Hash už zná, může spočítat hashe 0-1 a 0*
    - *Není potřeba si pamatovat všechny S0, S1*
        - *Stačí PRNG (inicializovaný klíčem), který vygeneruje potřebné S0, S1*
        - *Postupné generování*
    - *Spodní patro stromu není potřeba generovat najednou*
        - *Místo hashe se ve stromě použijí podpisy*
    - *Podpisů může být i nekonečno – SPHINC256*

24. *Co znamená "Perfect forward secrecy" jakým způsobem lze zajistit*
    - *Šifrovaná komunikace pomocí RSA/AES*
        - *Alice náhodně zvolá klíč symetrické šifry k*
        - *Alice pošle Bobovi RSA (k_{pubB}, k), AES (k,m)*
        - *Bob…*
        - *Po skončení účastníci zapomenou k*
    - *Nevýhoda – postrádá perfect forward secrecy*
        - *Útočník odposlechne komunikaci*
        - *Útočník v čase ukradne Bobovi jeho soukromý klíč k_{prB}*
        - *Útočník zpětně dešifruje k a následně m*
        - *Vyžaduje součinnost Boba – klíče a a b se generují pro každou relaci znovu, označují se jako „ephemeral" (efemérní, prchavý, dočasný)*

```
A                                                    B
                                                  B = g^b
                          <- sig                sigB = RSA(k_prB, B)
ver(sig, k_pub)
                      A, AES(k, m) ->
```

25. *Asymetrický systém DH:*
    - *K čemu slouží*
        - *Kryptografický protokol, který umožňuje přes nezabezpečený kanál vytvořit mezi komunikujícími stranami šifrované spojení, bez předchozího dohodnutí šifrovacího klíče – algoritmus pro výměnu klíčů*
    - *Postup volby parametrů*
        - *Prvočíslo p, generátor g z Zp*, kdy každé číslo od 1 do p může být generováno jako mocnina g*
        - *Alice si zvolí náhodné číslo a, 1<=a<=p-2 a vypočítá dle vzorce A*
        - *Bob si zvolí náhodné číslo a, 1<=b<=p-2 a vypočítá dle vzorce B*
        - *a ,b – soukromé klíče*
    - *Výpočet klíčů*

Alice → Bob: $A = g^a \bmod p$

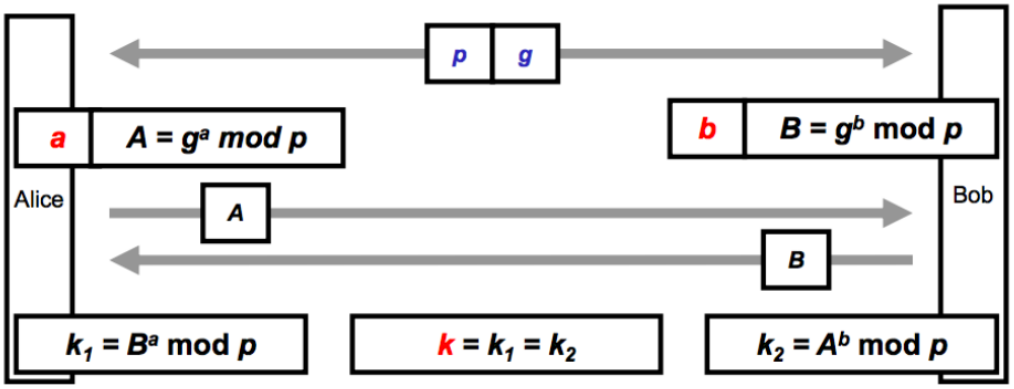
Bob → Alice: $B = g^b \bmod p$

- *Popis funkce protokolu*

Alice - vypočítá klíč k_1 jako $k_1 = B^a \bmod p$
$= (g^b \bmod p)^a \bmod p$
$= (g^b)^a \bmod p$

Bob - vypočítá k_2 jako $k_2 = A^b \bmod p$
$= (g^a \bmod p)^b \bmod p$
$= (g^a)^b \bmod p$

Obě strany došly ke stejnému číslu $k = k_1 = k_2$



26. *Asymetrický systém RSA:*
    - *Postup volby parametrů, výpočet klíčů*
        - *p a q volíme tak, aby každé z nich mělo ½ s bitů, typicky 1024<= s <= 4096*
        - *veřejný exponent e se volí jako 2^x+1 – čím menší tím rychlejší, ale zranitelnější*

p, q
n = pq
$\phi(n) = (p-1)(q-1)$
e < n, n nesoudělné s $\phi(n)$; často 65537 ($2^{16}+1$)
soukromý klíč d
ed = 1 mod $\phi(n)$, d = e^{-1}

    - *Postup pro šifrování a dešifrování*

Šifrovanie

$$c = m^e \bmod n$$

Dešifrovanie

$$m = c^d \bmod n$$

    - *Postup podepsání a ověření podpisu*
        - *Podepisuji soukromým klíčem, veřejným protistrana ověřuje*

27. *Slepý podpis na základě RSA: jak se vytváří, k čemu slouží?*
    - *Princip slepého podpisu*
        - *Autor zprávy a podepisující jsou různí, podepisujícímu je obsah zprávy utajen*
        - *Když se podepisující setká a odtajněnou zprávou, nesmí poznat, kdy a komu ji podepsal*
    - *Příklady*
        - *Elektronické volby (volební komisař podepisuje hlasovací lístky, ideálně neví ale jak jsme hlasovali)*
        - *Elektronické mince (vydavatel podepisuje mince, utajené jejich použití*
    - *Slepé podepsání zprávy m*
        - *Zvoláme náhodné číslo r < n*
        - *Vytvoříme zaslepenou zprávu $m' = mr^e \bmod n$*
        - *Předáme m' k podpisu a dostaneme $s' = m'^d \bmod n$*
        - *Spočítáme $s = s'r^{-1} \bmod n$*
        - *Platí $s = (mr^e)^d \equiv m^d \bmod n$*

28. *Asymetrický systém ElGamal:*
    - *Rozšíření Diffie-Hellmana*
    - *Šifrování*

Alice **Bob**

volba prvočísla: p

volba generátoru: $g \in Z_p^*$

volba soukromého klíče: $k_{pr}, d \in \{2, \dots, p-2\}$

výpočet veřejného klíče: $k_{pub}, e \equiv g^d \bmod p$

$k_{pub} = (e, g, p)$ ←

Volba: $k \in \{2, \dots, p-2\}$

výpočet „ephemeral" klíče (efemérního, dočasného):
$k_E \equiv g^k \bmod p$

výpočet maskovacího klíče: $k_M \equiv e^k \bmod p$

šifrování zprávy: $m \in Z_p^*$

$c \equiv m \cdot k_M \bmod p$

(k_E, c) →

výpočet maskovacího klíče: $k_M \equiv k_E^d \bmod p$

dešifrování: $m \equiv c \cdot k_M^{-1} \bmod p$

    - *Podpis*

Alice **Bob**

Vytvoření klíčů

volba prvočísla: p

volba generátoru: $g \in Z_p^*$

volba soukromého klíče: $k_{pr}, d \in \{2, \dots, p-2\}$

výpočet veřejného klíče: $k_{pub}, e \equiv g^d \bmod p$

veřejný klíš Boba k ověření podpisu

$k_{pub} = (e, g, p)$ ←

Vytvoření podpisu zprávy m

volba „ephemeral" klíče: $k_E \in \{2, \dots, p-2\}$ tak, aby platilo $\gcd(k_E, p-1) = 1$, (existuje inverzní prvek).

výpočet hodnot pro podpis zprávy:

$r \equiv g^{k_E} \bmod p$

Zpráva m s podpisem (r, s) $s \equiv (m - d \cdot r) k_E^{-1} \bmod p - 1$

$m(r, s)$ ←

Ověření podpisu

$k_E \equiv g^k \bmod p$

výpočet hodnoty: $t \equiv e^r \cdot r^s \bmod p$

ověření: $t \equiv g^m \bmod p$ - podpis validní

$t \not\equiv g^m \bmod p$ - podpis neplatný

29. *Asymetrický systém ECDH nad Fp: k čemu slouží*
    - *Analogický s klasickým D-H algoritmem*
    - *Umožňuje komunikujícím stranám získat sdílenou tajnou informaci jako například klíč pro klasickou symetrickou šifru*
    - *Daná je eliptická křivka definovaná nad Fp a pod P na eliptické křivce, který je generátor cyklické grupy*

- **Soukromý klíč: Alici číslo $k_{prvA} = a$, Bob číslo $k_{prvB} = b$**
- **Alice:** veřejný klíč $k_{pubA} = A = a \cdot P$
- **Bob:** veřejný klíč $k_{pubB} = A = b \cdot P$
- Alice s Bobem si vymění A a B,
- Alice $K_{AB} = a \cdot B$ Bob $K_{AB} = b \cdot A$
- $K_{AB} = a \cdot (b \cdot P) = b \cdot (a \cdot P)$ možné použít jako klíč pro symetrickou šifru; K_{AB} – souřadnice bodu na křivce
- útočník může zachytit pouze ***P, a·P, b·P***

30. *Asymetrický systém ECDSA nad Fp:*
    - *Slouží na podepisování a ověřování podpisu*
    - *Není možné použít k šifrování*
    - *Analogický k DSA*
    - *První základní algoritmus vytvářející EC*
    - *3 základní kroky*
        - *Generování klíče*
        - *Generování podpisu*
        - *Ověření podpisu*

31. *Pomocí čeho lze ověřit identitu určité entity? Uveďte příklady autentizačních metod.*
    - *Autentizace uživatelů*
        - *Znalost – login/heslo, výzva/odpověď, zero-knowledge*
        - *Předmět – úložiště s nechráněnými/chráněnými daty*
        - *Biometrika – fyziologie, behaviorální metody*
    - *Jedno a více faktorová autentizace*
    - *Jednostranná, vzájemná, jednorázová, opakovaná*

32. *Autentizace znalostí, stručně charakterizujte, uveďte používané metody.*
    - *Login – heslo*
        - *10+ znaků, mimo slovník, plná ASCII tabulka*
        - *Hashováno, šifrováno, u uživatelenic*
    - *Výzva odpověď*
        - *Žádost + login*
        - *Nonce*
        - *Hash hesla a nonce = HMAC*
        - *Odpověď*
    - *Nulová znalost*
        - *Ověřovatel může dokázat druhé straně, že dané tvrzení je pravdivé aniž by sdělovala jakoukoliv jinou informaci kromě skutečnosti, že tvrzení je skutečně pravdivé*

33. *Princip autentizace účastníků GSM sítě, používané algoritmy.*
    - *Global systém for mobile communication*
    - *Autentizace na základě SIM (subscriber identity module) – obsahuje 128b klíč sdílený s operátorem*
    - *Autentizace typu výzva-odpověď*
    - *Anonymitu zajišťuje TMSI (temporary mobile subscriber identity)*
        - *Dočasný identifikátor v rámci celé sítě*
        - *Snižuje možnost monitorování a sledování účastníků*
    - *Šifry*
        - *A3 – autentizace v SIM, vypočítává SRES*
        - *A8 – generování klíče pro A5 v SIM*

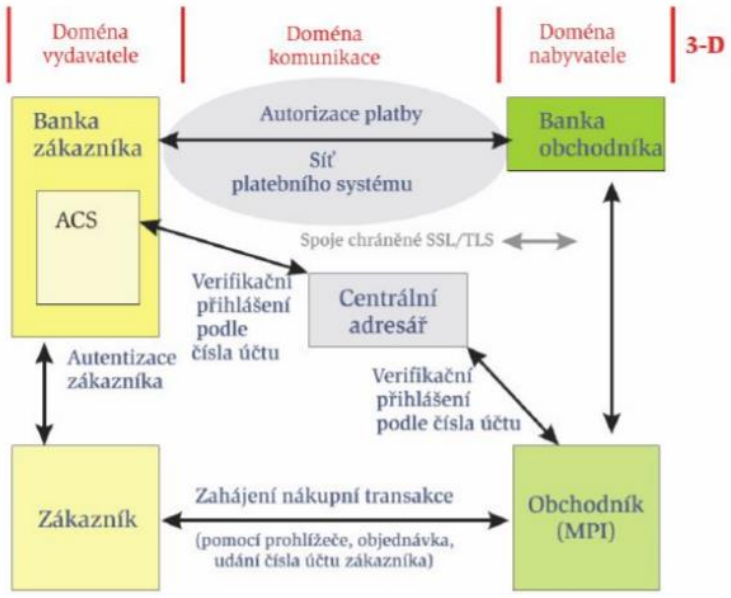
- *A5 – šifrování v MS*
  - *Není zabezpečená autentizace sítě a integrita*
  - *A5/0 – bez šifrování*
  - *A5/1 – proudový šifrovací algoritmus 64/54b klíč (64b z toho 10 je nulových, efektivní délka tedy je 54b)*
  - *A5/2 – oslabená verze A5/1*
  - *A5/3 – algoritmus používaný v UMTS (algoritmus Kasumi)*
- *Algoritmy*
  - *f0*
  - *f1 – autentizace sítě*
  - *f1* - na obnovu synchronizace autenticity zpráv*
  - *f2 – autentizace uživatele*
  - *f3 – generování šifrovaného klíče*
  - *f4 – generování klíče integrity*
  - *f5 – generování klíče anonymity*
  - *f5* - resynchronizace klíče anonymity*
  - *f8 – algoritmus pro šifrování*
  - *f9 – algoritmus pro integritu*
  - *Kasumi – Feistelova konstrukce – 8 rund*

34. *Co jsou to postranní kanály, uveďte a charakterizujte jejich základní principy a typu.*
    - *Každý nežádoucí způsob výměny informace mezi kryptografickým modulem a jeho okolím*
    - *Algoritmy jsou bezpečné, útočí se na implementace*
    - *Analýza postranního kanálu – postup získání užitečné informace ze signálu*
    - *Útok postranním kanálem – využití takové informace k napadení kryptografického modulu*
    - *Časová analýza*
      - *Měřením času jednoho průchodu lze zjistit správný znak na dané pozici*
      - *Místo 190 let lze na řešení přijít do půl hodiny (256*8 možností) -> obrana konstantní časovou implementací*
    - *Proudová analýza*
      - *Představuje efektivní a úspěšný způsob útoku na bezpečné algoritmy typu AES*
      - *Je přímo závislá na probíhajících operacích a zpracovávaných datech*
    - *Elektromagnetická analýza*
      - *Demonstrováno na extrakci obrazu z CRT monitoru*
      - *Spotřeba některých čipů odpovídá Hemingově váze (počet jedniček/vzdálenosti)*
    - *Akustická analýza*
      - *Už od studené války – ze zvuku mikroprocesoru v PC lze zjistit jednotlivé operace*
    - *Optická analýza*
      - *Paměťová buňka čipu „blikla“ při změně stavu*

35. *Načrtněte a popište strukturu blockchainu.*
    - *Je založen na řetězení bloků pomocí hashe bloku*

- *Každý blok obsahuje*
  - *Hash obsahu celého bloku*
  - *Data (seznam transakcí, hodnota nonce, timestamp...)*
  - *Hash předchozího bloku*

36. *Platební systém 3D Secure, vysvětlete základní princip činnosti.*
    - *Karetní asociace Visa, MasterCard*
      - *Jednotný systém pro zúčtování karetních transakcí na internetu*
        - *Číslo karty se předává přímo bance obchodníka*
      - *Zabezpečení pomocí SSL/TSL*
    - *Tři domény*
      - *Issure Domain – doména vydavatele – vydavatel kreditní karty je zodpovědný za přihlášení a ověření zákazníka během online transakce, který vlastní jejich kartu*
      - *Acquirer Domain – doména nabyvatele – zajišťuje aktuální zpracování a ověření, zda je obchodník zapojen do systému 3-D secure*
      - *Interoperability Domain – doména komunikace – ulehčuje transakci i přes běžně používané protokoly a sdílené systémy*



- *Výměna informací v systému 3-D Secure*
  - *Zákazník vyplní informace a předá je Merchant Server Plug-In na serveru*
  - *MPI je předá Directory Server (DS) pro kontrolu*
  - *DS zažádá příslušný Access Control Server (ACS) o ověření, že má zákazník registrovanou službu s podporou 3D Secure*
  - *ACD odpoví přes DS a MPI, zpráva obsahuje ACS URL*
  - *MPI posílá žádost o ověření zákazníka, kterého prohlížeč na URL přesměruje*
  - *Uživatel zadá Pin – tj. platbu autentizuje*
  - *ACS odešle podepsanou zprávu s výsledkem MPI*
  - *Výsledek ověření se zazálohuje*
  - *MPI ověří pravost podpisu a odešle žádost ověření ke své bance*
  - *Banka obchodníka žádá o ověření u vydavatele (banka)*
  - *Vydavatel potvrdí žádost a odešle odpověď*

37. *Protokol SSL, k čemu slouží, co zajišťuje, jaké kryptografické techniky využívá, naznačte, jakým způsobem pracuje.*
    - *Nalezneme mezi transportní a aplikační vrstvou*
    - *Umožňuje autentizaci mezi serverem a klientem*
    - *Autentizace serveru*
      - *Ověření identity*
      - *Klient vyžaduje autentizaci serveru, klientovi se zašle certifikát, na kterém je aby se řádně autentizoval*
      - *Ověření doby platnosti certifikátu, je vydávající CA důvěryhodnou?, lze veřejným klíčem CA ověřit pravost podpisu vydavatele?, odpovídá doméno jméno na certifikátu skutečnému jménu serveru?*
    - *Autentizace klienta*
      - *Málo používané*
    - *Šifrované spojení*
    - *Integrita dat pomocí MAC*
    - *Integrita toku zpráv je chráněna pořadovým číslem*
    - *Ochrana komunikace typu end-to-end*
    - *Nejrozšířenější aplikace používají http – jinak například SMTP,POP3, FTP…*

38. *Uveďte ideový návrh kryptosystému:*
    - *pro autentizaci uživatelů přistupujících vzdáleně k serverové aplikaci. Popište kryptografické služby, které systém musí zajišťovat, uveďte kryptografické algoritmy, které lze využít.*
    - *pro důvěrnou komunikaci mezi senzory a vzdáleným řídicím systémem. Popište kryptografické služby, které systém musí zajišťovat, uveďte kryptografické algoritmy, které lze využít.*
    - *zajišťující důvěrný přenos multimediálních dat v reálném čase, jednotlivé stanice mají dostateční výpočetní výkon a dostatečný paměťový prostor. Popište kryptografické služby, které systém musí zajišťovat, uveďte kryptografické algoritmy, které lze využít.*