Securitatea Rețelelor și Serviciilor de Comunicații
Criptografie. Probleme recapitulative.

1. Confidențialitatea datelor - Scheme de criptare 2
1.1. Confidențialitate. Email. 2
1.2. Modele de securitate. Scheme de criptare. 2
1.3. Criptare CBC. 2
1.4. Securitatea criptării CBC. IV fix sau predictibil (contor). 3
1.5. Criptare CTR. 3
1.6. Criptare folosind RSA. Securitate. 3
1.7. Scheme bazate pe logaritmul discret 4
2. Autentificarea datelor 5
2.1. Funcții hash criptografice. Cerinte de securitate. 5
2.2. Coduri de autentificare (MAC) și semnături digitale. Modele de securitate. 5
2.3. Cod de autentificare a mesajelor cu cifru bloc. CBC-MAC, CMAC. 5
2.4. Cod de autentificare a mesajelor cu funcții hash. HMAC. 6
2.5. Semnături digitale. Atacuri asupra funcției hash. 6
2.6. Semnătura digitală folosind RSA. 6
3. Protocoale de autentificare și stabilire a cheilor 7
3.1. Protocoale AKE folosind PSK and MAC (+ PFS/DH). 7
3.2. Protocoale AKE folosind SIG + DH. 7

1. Confidențialitatea datelor - Scheme de criptare

1.1. Confidențialitate. Email.

Alice trebuie să-i trimită lui Bob un fișier care conține un document confidențial. Alice știe că Bob folosește AES în mod CTR cu cheie de 128 de biți pentru criptare simetrică și RSA-OAEP cu cheie de 2048 de biți pentru criptare asimetrică. De asemenea, Alice are o copie a cheii lui Bob pentru criptare RSA.

Alice are instalate pe calculatorul său aplicații criptografice care implementează aceste scheme de criptare, dar Bob nu i-a explicat cum să cripteze documentul. Alice este nedumerită: RSA nu pare să fie soluția potrivită pentru criptarea fișierului, iar Bob nu i-a dat și o cheie pentru AES.

Ajutați-o pe Alice: Explicați cum ar trebui să cripteze fișierul și cum poate Bob să îl decripteze (fără detalii privind AES-CTR și RSA). Justificați soluția aleasă (ce operații trebuie efectuate și de ce).

Indicații:

Schemă de criptare hibridă.

1.2. Modele de securitate. Scheme de criptare.

a). Definiți securitatea perfectă (necondiționata) pentru scheme de criptare (perfect secrecy).

b). De ce o schemă de criptare cu cheie publică nu poate să ofere securitate perfectă?

c). Definiți cerința de securitate IND-CPA (definiți CPA și apoi definiți IND-CPA pe baza experimentului).

d). Demonstrați că schema de criptare cu cifru bloc ECB nu îndeplinește cerința de securitate IND-CPA.

Indicații:

b). În definiția securității perfecte (necondiționate) se consideră un adversar cu resurse (de calcul) nelimitate. Se poate argumenta în mai multe feluri că această noțiune de securitate nu este compatibilă cu criptarea cu cheie publică: de pildă, pornind de la (1) adversarul cunoaște funcția de criptare sau (2) criptarea cu cheie publică se bazează pe o permutare cu sens unic. [...]

d). Adversarul câștigă jocul IND-CPA cu probabilitate 1 după o singură interogare (vezi curs).

1.3. Criptare CBC.

a). Descrieți schema de criptare cu cifru bloc CBC.

b). Explicați securitatea oferită de criptarea CBC: ce noțiune de securitate îndeplinește această schemă de criptare și în ce condiții?

c). Schema CBC este vulnerabilă la atacuri în care adversarul găsește într-o secvență de blocuri criptate cu aceeași cheie, c_1, ..., c_m, două blocuri identice, $c_i = c_j$ ("matching ciphertexts").

Explicați cum se pot obține informații despre datele transmise (blocuri de text clar) printr-un astfel de atac și care este complexitatea atacului (i.e., câte blocuri cifrate trebuie să intercepteze adversarul pentru a găsi cu probabilitate mare două blocuri identice). Indicați o metodă practică pentru evitarea unor astfel de atacuri.

d). Demonstrați că CBC este o schemă de criptare cu auto-sincronizare (self-sinchronizing).

Indicații:

a). Vezi curs, CBC: Algoritmul de criptare CBC și algoritmul de decriptare.

b). Vezi curs, CBC: Condițiile pentru ca CBC să îndeplinească cerința de securitate IND-CPA.

c). In CBC, $c_i = E_K(m_i \oplus c_{i-1})$. This implies that $E_K(m_i \oplus c_{i-1}) = E_K(m_j \oplus c_{j-1})$, hence $m_i \oplus c_{i-1} = m_j \oplus c_{j-1}$ (since E_K is an injective function), so the adversary can compute $m_i \oplus m_j = c_{i-1} \oplus c_{j-1}$ (a function of plaintexts which does not depend on the secret key). This exposes the plaintext blocks; e.g., knowing m_i the adversary can compute m_j. Assuming that $F_K(m_i, c_{i-1}) = E_K(m_i \oplus c_{j-1})$ behaves like a pseudorandom function, an adversary can find two blocks $c_i = c_j$ after capturing about $2^{n/2}$ blocks (birthday paradox). To avoid this attack, the key must be changed after encrypting $t << 2^{n/2}$ blocks.

d). Dacă se pierd porțiuni de text cifrat, decriptarea CBC se restabilește automat după 2 blocuri cifrate consecutive: $m_j = c_{j-1} \oplus D_K(c_j)$.

1.4. Securitatea criptării CBC. IV fix sau predictibil (contor).

Demonstrați că variantele schemei CBC listate mai jos nu oferă securitate IND-CPA, descriind cum poate adversarul să câștige jocul IND-CPA:

a). Schemă de criptare CBC cu IV fixat.

b). Schemă de criptare CBC cu valori IV obținute folosind un contor.

Indicații:

Vezi curs: descrierea experimentului IND-CPA.

Notăm E_K^{CBC} funcția de criptare CBC, E_K funcția de criptare a cifrului bloc, n lungimea blocului, b bitul ales aleator de oracol la inițializare.

a). CBC cu IV fixat, IV = v.

Mai notăm $c_0 = E_K^{CBC}(0^n) = E_K(v \oplus 0^n)$ și $c_1 = E_K^{CBC}(1^n) = E_K(v \oplus 1^n)$; funcția E_K este injectivă deci $c_0 \neq c_1$. Adversarul câștigă după q = 1 interogări, cu probabilitate 1, exploatând faptul că pentru un IV fixat algoritmul de criptare este determinist. De exemplu:

- Adversarul află de la oracol $c_0 = E_K^{CBC}(0^n)$.
- Adversarul trimte interogarea ($m_0 = 0^n$, $m_1 = 1^n$) și oracolul răspunde cu provocarea $c = E_K^{CBC}(m_b)$. Dacă b = 0, atunci $c = E_K^{CBC}(0^n) = c_0$, altfel $c = E_K^{CBC}(1^n) = c_1$. Prin urmare, adversarul răspunde b = 0 dacă $c = c_0$, altfel răspunde b = 1.

b). CBC cu IV = contor mod 2^n.

Experimentul IND-CPA prezentat în curs poate fi adaptat pentru scheme care folosesc vector de inițializare: oracolul de criptare raspunde la interogări cu (IV, c), c este textul criptat, iar IV este vectorul de inițializare folosit pentru a calcula c. Presupunem că valoarea curentă a contorului IV este v și adversarul o cunoaște (din algoritm sau printr-o interogare). Adversarul câștigă după q = 1 interogări, cu probabilitate 1, exploatând faptul că poate anticipa următoarea valoare IV. De exemplu:

- Adversarul află de la oracol $c_0 = E_K^{CBC}(v)$ $(v, E_K(v \oplus v)) = (v, E(0^n))$. Pentru următoarea criptare IV = v +1 mod 2^n; pentru a simplifca notația presupunem v +1 mod 2^n = v +1.
- Adversarul trimte interogarea ($m_0 = v+1$, m_1), $m_1 \in \{0,1\}^n$, $m_1 \neq m_0$. Oracolul răspunde cu provocarea $c = (v+1, E_K^{CBC}(m_b))$. Dacă b = 0, $c = E_K^{CBC}(v+1) = (v+1, E_K(0^n)) = c_0$, altfel $c = E_K^{CBC}(m_1) \neq c_0$. Prin urmare, adversarul răspunde b = 0 dacă $c = c_0$, altfel răspunde b = 1.

1.5. Criptare CTR.

a). Descrieți schema de criptare cu cifru bloc CTR.

b). Explicați cum se pot obține informații despre datele transmise (blocuri de text clar) dacă o implementare a schemei de criptare CTR repetă valori ale contorului pe durata de viață a unei chei.

c). Explicați securitatea oferită de criptarea CTR: ce noțiune de securitate îndeplinește această schemă de criptare și în ce condiții?

d). Explicați cum putem extinde schema de criptare CTR pentru a obține o schemă de criptare IND-CCA.

Indicații:

a). Vezi curs, CTR: Algoritmul de criptare și algoritmul de decriptare.

b). Notăm X_i valoarea contorului și E_K funcția de criptare a cifrului bloc. Dacă $c_i = m_i \oplus E_K(X_i)$ and $c_j = m_j \oplus E_K(X_i)$, atunci $c_i \oplus c_j = m_i \oplus m_j$. Aflând m_i, adversarul poate calcula $m_j = c_i \oplus c_j \oplus m_i$.

c). Vezi curs, CTR: Condițiile pentru securitate IND-CPA.

d). Folosind criptare autentificată: Criptare cu CTR și autentificare cu MAC (encypt then MAC): $E_{K1}(m)$, $MAC_{K2}(E_{K1}(m))$.

1.6. Criptare folosind RSA. Securitate.

a). Definiți funcția RSA și justificați ipoteza că este o permutare cu sens unic și trapă (trapdoor one-way permutation). (Arătați că este (1) permutare (funcție bijectiva), (2) funcție cu sens unic și (3) cu trapa.)

b). Pornind de la definiția generică unei scheme de criptare asimetrică, definiți o schemă care folosește funcția RSA și o codare deterministă a textului clar (de exemplu, completare cu 00 ... 01).

c). Explicați de ce schema definită la punctul precedent nu poate să îndeplinească cerințele uzuale de securitate ale schemelor de criptare (de exemplu IND-CPA).

d). Explicați cum putem obține o schemă de criptare bazată pe funcția RSA care poate oferi securitate IND-CCA (principiu, fără detalii privind algoritmul de codare).

e). Descrieți atacul pentru recuperarea cheii prin forță brută în cazul schemelor criptografice bazate pe RSA.

Indicații:

Vezi curs.

a). The RSA function is a trapdoor one-way permutation.

(1) Permutation: Let n = pq, with p, q primes.

$RSA_{n,e} : Z_n \to Z_n$, where $gcd(e, \varphi(n)) = 1$ and $RSA_{n,e}(m) = m^e \bmod n$ for $m \in Z_n$.

Let $RSA_{n,d} : Z_n \to Z_n$, where $d = e^{-1} \bmod \varphi(n)$ and $RSA_{n,d}(c) = c^d \bmod n$ for $c \in Z_n$.

One can (easily) prove that $RSA_{n,d}$ is the inverse of $RSA_{n,e}$. This implies that RSA is a permutation.

(2) One-way: Define one-way functions and show that RSA satisfies the two requirements. Knowing e and n, one can efficiently compute $RSA_{n,e} = m^e \bmod n$ (polynomial complexity). The best known way to determine $RSA_{n,d} = RSA^{-1}_{n,e}$ is to find the factors of n and then compute $d = e^{-1} \bmod \varphi(n)$. Factoring n is infeasible for large n (sub-exponential complexity).

(3) Explain what is the trapdoor that allows efficient computation of $RSA^{-1}_{n,e}$ and how/why this trapdoor can be kept secret. The trapdoor is the exponent $d = e^{-1} \bmod \varphi(n)$. As explained above, finding d given e and n is assumed to be as hard as factoring n.

b). Adapt the material presented in the lecture slides.

c). The encryption algorithm is deterministic and stateless.

d). See the RSA assumption and the generic RSA-based probabilistic encryption in the lecture slides.

e). Start from the basic security requirements. How do we keep the trapdoor secret?

1.7. Scheme bazate pe logaritmul discret.

Considerăm un grup ciclic G de ordin prim q în care problema DDH (Decisional Diffie-Hellman) este dificilă (intractabilă). Fie g un generator al acestui grup și $f : Z^*_q \to G$, $f(x) = g^x$ funcția exponențială discretă în G.

a). Justificați afirmația că funcția definită mai sus este o permutare cu sens unic.

(Explicați succesiv de ce este (1) o permutare (funcție bijectiva), (2) funcție cu sens unic.)

b). Descrieți cum poate fi folosită această funcție pentru a obține un protocol de distribuție a cheilor secrete. Explicați securitatea oferită de acest protocol (cerințe de securitate, securitatea oferită de protocol, pentru care tipuri de atacuri și în ce condiții - ipoteze asupra parametrilor).

c). Care este principalul parametru de securitate al schemelor criptografice bazate pe logaritmul discret? Explicați cum afectează acest parametru securitatea acestor scheme.

Indicații:

Vezi materialul de curs.

a). (1) Permutare: Prin construcție, având în vedere proprietățile grupului (ciclic). (2) Sens unic: Definiți problema DDH și explicați relația cu problema DL. Explicați de ce și în ce condiții se consideră că problema DL este intractabilă.

b). Diffie-Hellman (DH) key agreement.

Cerința de securitate: Un adversar (cu resurse limitate) care cunoaște g^x și g^y nu poate distinge g^{xy} de un element ales aleator uniform în G, dacă x și y sunt alese aleator uniform în G. Protocolul DH îndeplinește această cerință de securitate dacă G este un grup în care problema DDH este intractabilă, însă doar pentru atacuri pasive. Este vulnerabil la atacuri active de tip MITM (Man-In-The-Middle). Această vulnerabilitate poate fi eliminată folosind o variantă a protocolului DH cu autentificare mutuală.

c). Principalul parametru de securitate: ordinul q al grupului G (de obicei, lungimea reprezentării sale binare). Acest parametru determină complexitatea algoritmului de calcul al logaritmului discret, care reprezintă principalul atac prin forță brută asupra acestor scheme (complexitatea algoritmului depinde însă și de modul în care este definit grupul G). Pentru scheme în care G este un subgrup de ordin prim q al grupului Z^*_p, cu p prim, intervin 2 parametri: lungimea reprezentării binare a lui q (lungimea cheii private) și lungimea reprezentării binare a lui p (lungimea cheii publice).

2. Autentificarea datelor

2.1. Funcții hash criptografice. Cerinte de securitate.

Notăm $H : X \to Y$, cu $X = \{0,1\}^*$ și $Y = \{0,1\}^n$, o funcție hash criptografică.

a). Definiți proprietățile de securitate ale unei astfel de funcții: rezistența la preimagine, la a doua preimagine, la coliziuni (preimage, second preimage, collision resistance).

b). Descrieți atacul prin forță brută pentru fiecare proprietate de securitate și evaluați complexitatea sa.

c). Arătați că rezistența la coliziuni implică rezistența la preimagine și rezistența la a doua preimagine.

Indicații:

a). [...]

b). Adversarul alege în mod aleator $x \in X$ și calculează $H(x)$ până când găsește valorile relevante pentru proprietatea respectivă. Complexitatea atacurilor se deduce ca în cele două situații discutate la paradoxul zilei de naștere (birthday paradox).

Se folosesc proprietățile următoare: O funcție hash "comprimă" mulțimea intrărilor, în sensul că $|X| / |Y| >> 1$. De asemenea, ea distribuie uniform elementele din X către cele din Y, astfel încât $|H^{-1}(y)| \approx |X| / |Y|$ pentru oricare $y \in Y$. Prin urmare, pentru oricare $y \in Y$, alegând aleator uniform $x \in X$, $Prob[H(x) = y] = 1 / |Y|$.

c). Arătați că având un algoritm care găsește a doua preimagine se poate construi un algoritm care găsește eficient coliziuni; similar pentru preimagine (ținând seama de proprietatea de compresie). Deci nu putem obține rezistența la coliziuni decât dacă sunt îndeplinite celelalte două proprietăți.

2.2. Coduri de autentificare (MAC) și semnături digitale. Modele de securitate.

a). Descrieți modelul general de specificare a cerințelor de securitate ale schemelor criptografice.

b). Arătați cum se aplică acest model pentru schemele de autentificare a datelor cu cheie secretă (MAC): obiective, atacuri, cerințe de securitate.

c). Definiți cerința uzuală de securitate pentru schemele MAC.

2.3. Cod de autentificare a mesajelor cu cifru bloc. CBC-MAC, CMAC.

a). Descrieți schema de autentificare a datelor cu cheie secretă CBC-MAC (diagramă și explicații).

b). În ce condiții îndeplinește schema CBC-MAC (de bază) cerința de securitate UF-CMA?

c). Adversarul interceptează un mesaj protejat folosind această schemă, $(m, MAC_K(m))$. Arătați cum poate genera o pereche $(m', MAC_K(m'))$ care să fie acceptată de algoritmul de verificare fără să cunoască cheia K.

d). Explicați cum este contracarat acest atac în schema CMAC.

Indicații:

a). Vezi curs.

b). Cifrul bloc este PRP și mesajele au lungime fixată.

c). Falsificare prin metoda XOR (XOR forgery). Exemplu în curs. Se poate generaliza pentru a obține falsuri prin concatenarea a două mesaje arbitrare cu MAC cunoscut.

d). CMAC adaugă o transformare de ieșire care folosește o cheie secretă diferită și are rolul de a bloca mecanismul de înlănțuire al schemei CBC-MAC. Vezi curs.

2.4. Cod de autentificare a mesajelor cu funcții hash. HMAC.

a). O posibilă construcție a unui cod de autentificare folosind funcții hash H este metoda prefixului secret: $MAC_K(m) = H(K \| m)$. Arătați că această construcție nu îndeplinește cerințele de securitate pentru MAC.

b). O altă construcție posibilă este metoda sufixului secret: $MAC_K(m) = H(m \| K)$. Explicați de ce nici această metodă nu oferă o soluție adecvată.

c). Comparați schema HMAC cu schemele precedente, explicați cum elimină deficiențele identificate și care este efortul de calcul suplimentar consumat în acest scop.

Indicații:

Arătați cum poate genera adversarul o pereche $(m', MAC_K(m'))$ acceptată de algoritmul de verificare fără să cunoască cheia K și care este complexitatea atacului.

a). Atac pasiv prin extinderea mesajului: Adversarul a interceptat un mesaj $(m, MAC_K(m))$ și poate genera imediat $(m_2 = m \| m_1, MAC_K(m_2) = H(K, m \| m_1))$, pentru orice m_1. Simplu atac KMA.

b), Atac activ bazat pe coliziuni ale funcției hash: Presupunem că adversarul poate efectua atacuri CMA. Adversarul calculează offline m, m' astfel încat H(m) = H(m'). Aflând $(m, MAC_K(m))$ prin CMA, adversarul obține imediat și $(m', MAC_K(m') = MAC_K(m))$. Pentru o funcție H cu ieșire de n biți găsirea unei coliziuni (prin forță brută) necesită doar $2^{n/2}$ operații (evaluări ale funcției H). Pentru o schemă MAC cu cod de n biți și cheie de $k \geq n$ biți, cerința de securitate este ca nici un atac să nu aibă complexitate mai mică decât 2^n operații. De asemenea, rezistența la coliziuni este cerința cea mai dificilă și mai fragilă, prima care cedează la atacuri criptanalitice (vezi MD5, SHA1). Este necesară deci o construcție MAC care să fie mai puțin expusă la astfel de atacuri (vezi HMAC).

c). În esență, HMAC este o variantă a metodei prefixului secret la care este adăugată o transformare de ieșire care previne atacurile prin extinderea mesajelor. Eficiență: Luați în considerare posibilitatea de a pre-calcula unele componente.

2.5. Semnături digitale. Atacuri asupra funcției hash.

Descrieți cum pot fi falsificate mesaje protejate folosind o schemă de semnătură digitală bazată pe construcția hash-and-sign fără a cunoaște cheia privată, în scenariile de mai jos. Evaluați complexitatea fiecărui atac.

a). Eve interceptează un mesaj semnat de Alice și vrea să-l modifice înainte de a-l livra la destinație.

b). Eve redactează un contract pe care urmează să îl încheie cu Alice și vrea să obțină un contract semnat de Alice care prevede că Eve să primească o sumă dublă față de cea acceptată.

Indicații:

a). Atac asupra rezistenței la a doua preimagine a funcției hash. Vezi curs.

b). Atac asupra rezistenței la coliziuni a funcției hash. Vezi curs.

2.6. Semnătura digitală folosind RSA.

a). Descrieți principiul schemei de semnătură digitală realizată folosind: familia de funcții RSA, o funcție hash criptografică H și o funcție de codare P (diagramă și explicații).

b). În ce condiții îndeplinește schema de semnătură RSA-PSS cerința de securitate UF-CMA?

c). Eve cunoaște cheia publică folosită de Bob pentru o schemă de semnătură digitală bazată pe RSA și a obținut un mesaj semnat de Bob. Explicați cum ar putea încerca Eve să găsească cheia privată a lui Bob și cum este protejată schema de semnătură digitală împotriva unor astfel de atacuri.

Indicații:

a). Vezi curs (semnătură RSA cu codare).

b). Vezi curs: ipoteza RSA, funcție hash rezistentă la coliziuni.

c). Atac asupra funcției RSA (factorizarea modulului, etc.).

3. Protocoale de autentificare și stabilire a cheilor

3.1. Protocoale AKE folosind PSK and MAC (+ PFS/DH).

Alice (A) inițiază o comunicație cu serverul Bob (B) folosind protocolul KE-MA-MAC descris mai jos.

M0: $A \rightarrow B$: A, N_A M1: $B \rightarrow A$: $B, N_B, MAC_{K1}(B \| N_B \| N_A)$ M2: $A \rightarrow B$: $MAC_{K1}(A \| N_A \| N_B)$ Chei: $K_S = PRF_{K2}(N_A \| N_B)$.	Notații: N_A, N_B: valori aleatoare de 256 biți (random nonce). $MAC_K(D)$: cod de autentificare a datelor D cu cheia K. K0 = (K1, K2): cheie secretă pre-stabilită (PSK) între A și B. K1 este folosită pentru autentificare în protocol, iar K2 pentru a genera cheile de sesiune K_S (criptare, MAC), folosind PRF și N_A, N_B.

a). Explicați ce dovezi obține fiecare participant că cheia de sesiune stabilită (Ks) este cunoscută doar de celălalt participant. Este protocolul KE-MA-MAC vulnerabil la atacurile uzuale (repetare, reflectare, intercalare)? Oferă acest protocol confirmarea și indepedența cheilor (de sesiune)? Explicați.

b). Explicați de ce protocolul KE-MA-MAC nu oferă PFS. Definiți un protocol pentru distribuire autentificată a cheilor, KE-MA-MAC-PFS, care oferă PFS, prin modificari minime ale protocolului KE-MA-MAC.

Indicații:

a). Vezi curs, protocolul KE-PSK-MAC-2.

În faza de înregistrare, entitățile A și B asociază cheile secrete K1 și K2 identităților lor. Aceste chei nu mai sunt cunoscute de nici o altă entitate.

Autentificare: Vezi curs, protocolul MA-MAC. A și B se autentifică mutual prin demonstrarea cunoașterii cheii secrete K1, calculând $MAC_{K1}(A \| N_A \| N_B)$, respectiv $MAC_{K1}(B \| N_B \| N_A)$. Fiecare verifică dacă a primit valoarea MAC corectă folosind copia proprie a cheii K1.

Validitatea codului MAC dovedește că entitatea care l-a calculat cunoaște K1. Adversarul nu poate calcula codul corect pentru o intrare arbitrară decât cu probabilitate neglijabilă, deoarece nu cunoaște K1, iar schema MAC îndeplinește UF-CMA.

De asemenea, adversarul nu poate afla răspunsul corect dintr-o altă sesiune (atac prin repetare, reflectare sau intercalare) decât cu probabilitate neglijabilă, deoarece: fiecare răspuns depinde atât de N_A, cât și de N_B; N_A și N_B sunt alese uniform aleator și independent în $[0, 2^{256}-1]$, deci probabilitatea să se repete este neglijabilă (2^{-128} pentru fiecare, 2^{-256} amândouă); răspunsurile sunt asimetrice (calculate diferit). Ilustrați prin exemple faptul că atacurile repetare, reflectare sau intercalare eșuează.

A poate deci conclude că răspunsul pe care l-a primit poate proveni doar de la entitatea B. Similar, B conclude că răspunsul primit poate proveni doar de la A.

Secretul cheii Ks: A și B calculează cheia secretă partajată $K_S = PRF_{K2}(N_A \| N_B)$, unde funcția PRF poate fi construită folosind MAC. Validitatea codurilor MAC primite dovedește atât identitatea fiecărei entități, cât și faptul că N_A a fost ales de A, iar N_B a fost ales de B (nu de adversar), deci putem presupune că sunt alese aleator și independent în $[0, 2^{256}-1]$, conform protocolului. Adversarul nu cunoaște K2 și nu poate determina ieșirea PRF pentru o valoare arbitrară a intrării decât cu probabilitate neglijabilă (2^{-n}, pentru ieșiri de n biți).

Indepedență Ks: Da, dacă N_A și N_B sunt alese uniform aleator și independent pentru fiecare sesiune.

Confirmare Ks: Nu. [...]

b). PFS: [...]. Protocolul KE-MA-MAC-PFS folosește protocolul Diffie-Hellman pentru stabilirea cheii secrete (DH autentificat cu MAC), în mod similar cu protocolul KE-DH-SIG prezentat în curs.

3.2. Protocoale AKE folosind SIG + DH.

Alice (A) inițiază o comunicație cu serverul Bob (B) folosind protocolul UA-SIG descris mai jos.

M0: $A \rightarrow B$: I want to talk M1: $B \rightarrow A$: N_B M2: $A \rightarrow B$: $PKC(A, pk_A), SIG_{skA}(N_B)$	Notații: N_B este o valoare aleatoare de 256 biți (random nonce), $SIG_{skA}(D)$ este semnătura entității A pentru datele D cu cheia privată sk_A, iar $PKC(A, pk_A)$ este certificatul pentru cheia publică de semnătură pk_A a lui A.

a). Explicați cum determină Bob identitatea utilizatorului cu care comunică:

- Ce dovezi obține Bob privind identitatea utilizatorului?
- Ce rol are certificatul?

b). Definiți un protocol de autentificare mutuala, MA-SIG, prin modificări minime ale protocolului UA-SIG:

- Listați secvența de mesaje și explicați notațiile.
- Explicați cum determină fiecare participant identitatea celuilalt.
- Este protocolul propus vulnerabil la atacurile uzuale (replay, reflection, interleaving)? Explicați.

c). Explicați de ce protocoalele UA-SIG și MA-SIG nu sunt suficiente pentru a controla accesul la serviciul oferit de (serverul) Bob. Propuneți o soluție generală.

d). Definiți un protocol de stabilire a cheilor cu autentificare mutuală, KE-DH-MA-SIG, care combină protocolul MA-SIG și protocolul Diffie-Hellman:

- Listați secvența de mesaje, arătați cum sunt calculate cheile și explicați notațiile introduse.
- Ce dovezi obține A că cheile obținute sunt cunoscute de B și numai de B (și invers)?
- Este protocolul propus vulnerabil la atacurile uzuale asupra protocoalelor AKE (replay, reflection, interleaving)? Explicați.
- Oferă acest protocol confirmarea cheilor, indepedența cheilor și PFS? Explicați.

Indicații:

a). Vezi curs, UA-SIG.

Certificatul PKC(A, pk_A) asociază identității A cheia publică pentru semnătură pk_A. Asocierea este garantată de autoritatea de certificare AC care a emis și semnat digital PKC(A, pk_A). AC trebuie să fie o autoritate de certificare în care B are încredere. Pentru autentificare, entitatea A trebuie să demonstreze cunoașterea cheii secrete sk_A asociate cheii pk_A, calculând semnătura $SIG_{skA}(N_B)$. Entitatea B verifică PKC(A, pk_A) (semnătura AC, intervalul de validitate, etc.) și apoi verifică semnătura $SIG_{skA}(N_B)$ folosind pk_A.

Validitatea certificatului PKC(A, pk_A) dovedește că pk_A este cheia publică a entității A. Prin urmare, doar entitatea A cunoaște cheia privată sk_A asociată cheii publice pk_A.

Validitatea semnăturii dovedește că entitatea care a calculat-o cunoaște sk_A. Adversarul nu poate calcula semnătura corectă decât cu probabilitate neglijabilă, deoarece nu cunoaște sk_A, iar schema de semnătură îndeplinește UF-CMA. De asemenea, adversarul nu poate afla răspunsul corect dintr-o sesiune anterioară (atac prin repetare) decât cu probabilitate neglijabilă, deoarece N_B este ales uniform aleator în $[0, 2^{256}-1]$, deci probabilitatea să se repete este neglijabilă ($= 2^{-128}$). B poate deci conclude că răspunsul primit poate proveni doar de la entitatea A.

b). Protocolul MA-SIG este listat mai jos:

M0: A → B: A, N_A M1: B → A: B, N_B, $SIG_{skB}(A \| N_B \| N_A)$, PKC(B, pk_B) M2: A → B: $SIG_{skA}(B \| N_A \| N_B)$, PKC(A, pk_A)	Notații: A și B sunt identitățile celor 2 participanți. Notațiile introduse pentru B sunt similare celor folosite pentru A.

Autentificare: Operații: vezi curs, MA-SIG + certificate. Verificarea identității: similar cu UA-SIG.

Atacuri: Adversarul nu poate afla răspunsul corect dintr-o altă sesiune (atac prin repetare, reflectare sau intercalare) decât cu probabilitate neglijabilă, deoarece: fiecare răspuns depinde atât de N_A, cât și de N_B; N_A și N_B sunt alese uniform aleator și independent în $[0, 2^{256}-1]$, deci probabilitatea să se repete este neglijabilă (2^{-128} pentru fiecare, 2^{-256} amândouă); răspunsurile sunt asimetrice (calculate diferit). Ilustrați prin exemple faptul că atacurile repetare, reflectare sau intercalare eșuează.

c). Trebuie protejată întreaga comunicație, pentru a preveni accesul neautorizat la date prin eavesdropping și session hijacking: canal securizat care asigură autentificarea și confidențialitatea datelor.

d). Protocolul KE-DH-MA-SIG este listat mai jos:

M0: A → B: A, g^x M1: B → A: B, g^y, $SIG_{skB}(A \| g^y \| g^x)$, PKC(B, pk_B) M2: A → B: $SIG_{skA}(B \| g^x \| g^y)$, PKC(A, pk_A) Chei: $K_S = PRF(g^{xy})$.	Notații suplimentare: Se folosește DH cu parametrii (G, g, q). x și y sunt exponenți DH aleși aleator uniform în 1, ..., q-1. PRF(S) este aici un generator pseudoaleator inițializat cu valoarea S (seed), utilizat pentru a genera cheile de sesiune K_S (criptare, autentificare).

(Variantă simplificată. Protocoalele folosite de obicei în practică folosesc și N_A, N_B.)

Explicații: [...]