Securitatea Rețelor și Serviciilor de Comunicații

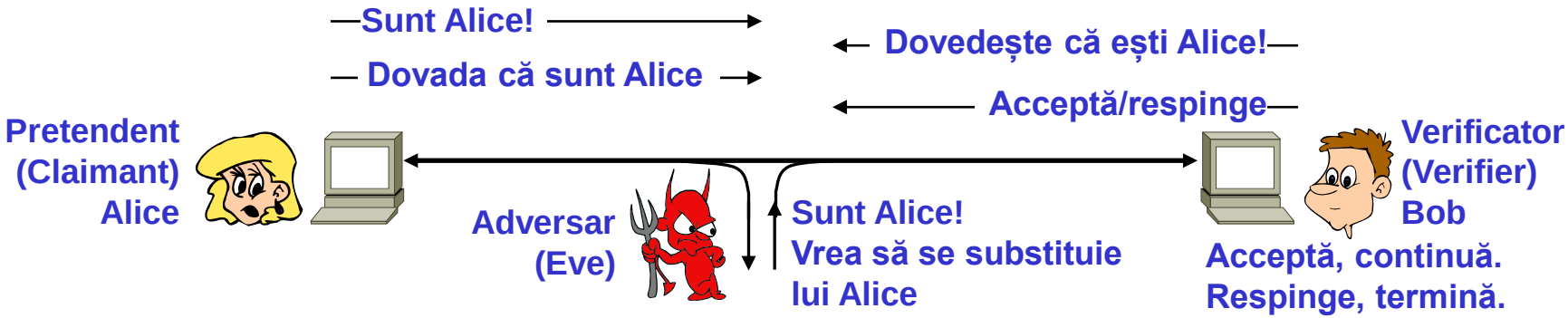
Protocoale criptografice: Protocoale de autentificare și de stabilire a cheilor

Protocoale criptografice

Protocoale de autentificare

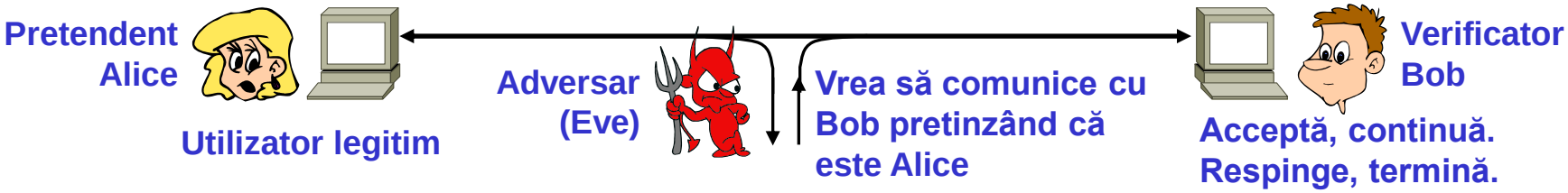
Autentificarea entităților

- *Proces interactiv* (*protocol de autentificare*) în care un participant (*verificator*) determină identitatea celuilalt participant (*pretendent*), pe baza unor *dovezi nefalsificabile*.
- *Autentificare unilaterală*: Un participant determină identitatea celuilalt.
- *Autentificare mutuală*: Fiecare participant determină identitatea celuilalt.



- ***Dovezi privind identitatea***
  - Ceva ce știe: un secret (exemple: parolă, PIN).
  - Ceva ce are: o dovadă fizică (exemple: cartelă).
  - Ceva ce este: date biometrice (exemplu: amprentă digitală).
  - Combinație a dovezilor de mai sus: autentificare multi-factor.

Model de securitate



Obiectivele adversarului

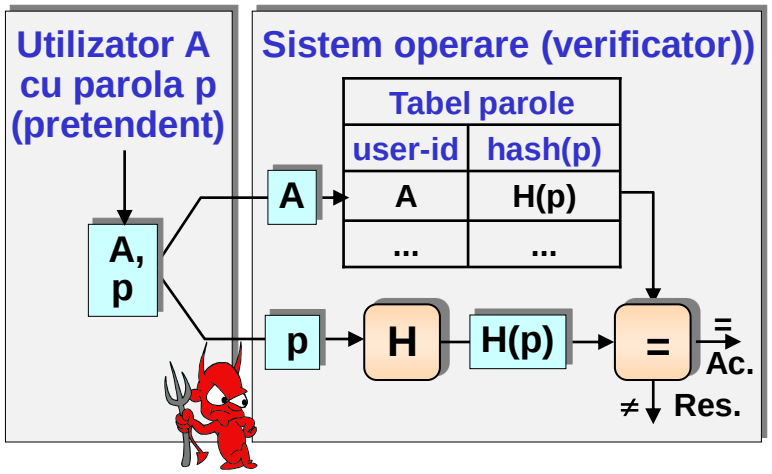
- Adversarul dorește să își asume o identitate falsă (a altei entități) în comunicații cu verificatorul. Ex: pretinde identitatea unei entități legitime pentru a putea utiliza drepturile de acces ale acesteia.

Capabilitățile adversarului

- Cunoaște algoritmii și protocoalele, nu cunoaște secretele.
- Adversar activ: Controlează comunicațiile dintre participanții legitimi: poate citi, modifica, șterge sau întârzia mesajele acestora, poate transmite mesaje proprii falsificând identitatea transmițătorului.
- Poate iniția sesiuni de autentificare, inclusiv sesiuni multiple în paralel.

Autentificare locală cu parolă

- Înregistrare: Asociază o identitate (A) unei parole (p) secrete (șir de caractere). Verificatorul stochează $H(p)$, unde H este o funcție hash criptografică. (De ce?)
- Autentificare: Pretendentul declară o identitate și demonstrează cunoașterea parolei secrete asociate acesteia.



Avantaje, vulnerabilități, atacuri, contramăsuri

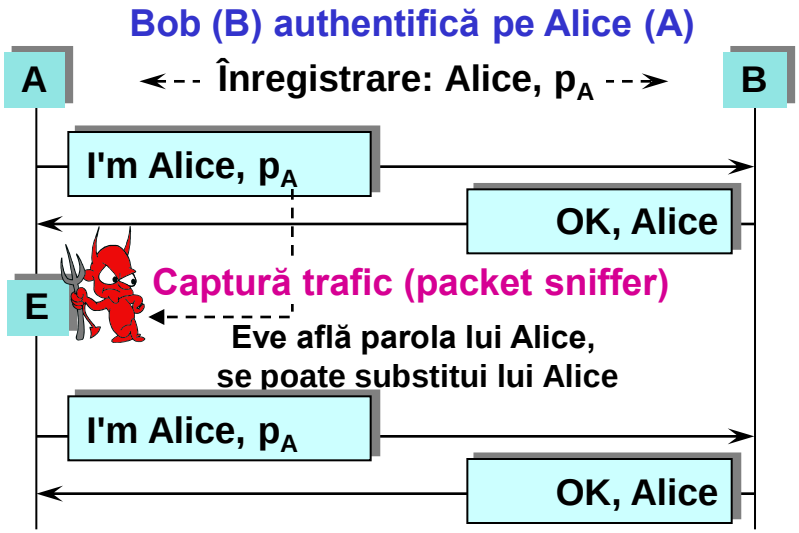
- Avantaj: Utilizatorul își poate alege o parolă mai ușor de memorat.
- Dezavantaj: Utilizatorul își alege o parola predictibilă, ușor de ghicit.
- "Dictionary attack": Adversarul verifică $H(p_i) = H(p)$ pentru o listă de parole uzuale $\{H(p_1), H(p_2), \ldots\}$. Poate fi executat offline.
- Contramăsuri: Parola este validată la înregistrare (lungime, caractere). Folosim o funcție H specială, cu durată mare de calcul. Extindem p cu un șir aleator, pentru a împiedica precalcularea $H(p_i)$.

Autentificare cu parole în rețea

Parole transmise prin rețea?

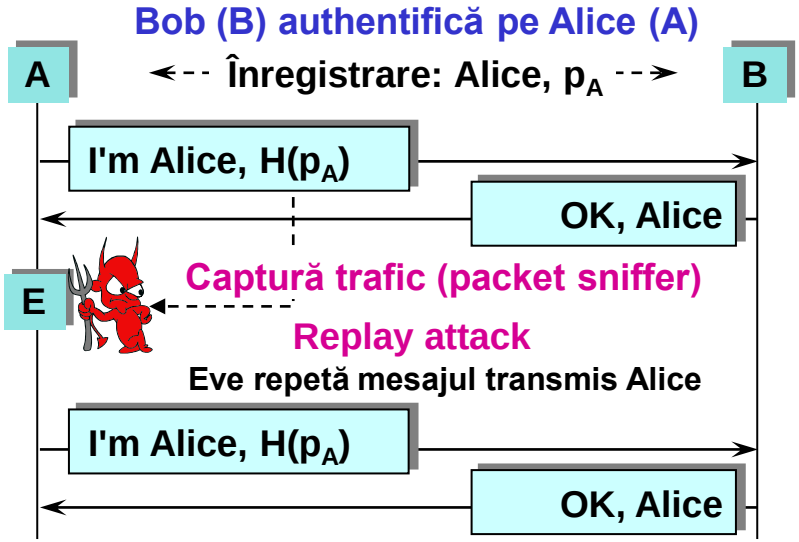
- Extindere a autentificării locale cu parolă pentru autentificare în rețea.

  Singurul protocol de autentificare disponibil inițial în Internet.
- *Mesajul dezvăluie secretul. Protocol vulnerabil la atacuri pasive: adversarul înregistrează traficul și află parola.*



Protocol cu parole criptate?

- Am putea transmite H(p) în loc de p, astfel încât adversarul să nu afle p.
- *Mesajul de autentificare poate fi repetat: Protocol vulnerabil la atacuri pasive prin repetare (replay attack).*



Protocoale provocare/răspuns

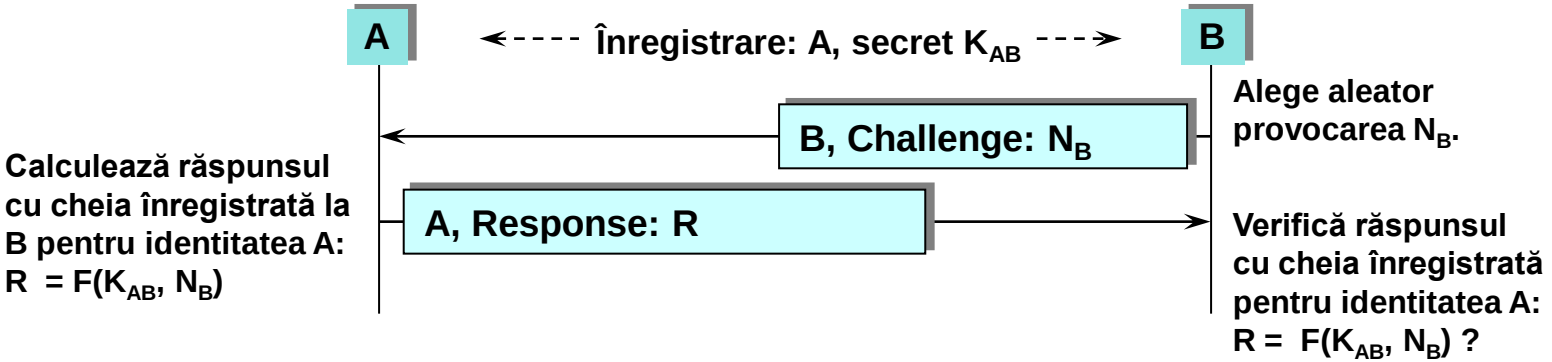
Obiectiv

- Pretendentul trebuie să demonstreze cunoașterea secretului asociat identității *fără a-l dezvălui*, printr-un dialog *nefalsificabil și nerepetabil*.

Protocoale cu provocare și răspuns (challenge-reponse)

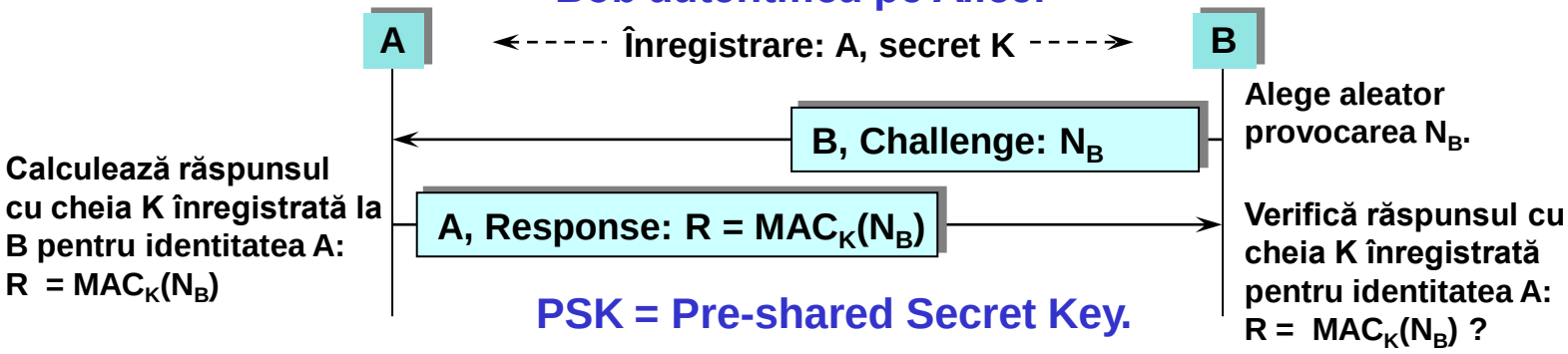
- Provocare trimisă de verificator: șir binar diferit la fiecare autentificare.
- Răspunsul pretendentului: funcție criptografică de provocare și de secretul asociat identității, care nu dezvăluie secretul.
- Ce funcții criptografice? Ce proprietăți trebuie să aibă provocarea?

Exemplu generic: Autentificare unilaterală cu secret prestabilit (K_{AB}). Bob autentifică pe Alice.



Autentif. unilaterală cu MAC

UA-MAC: Autentificare unilaterală cu PSK și MAC.
Bob autentifică pe Alice.

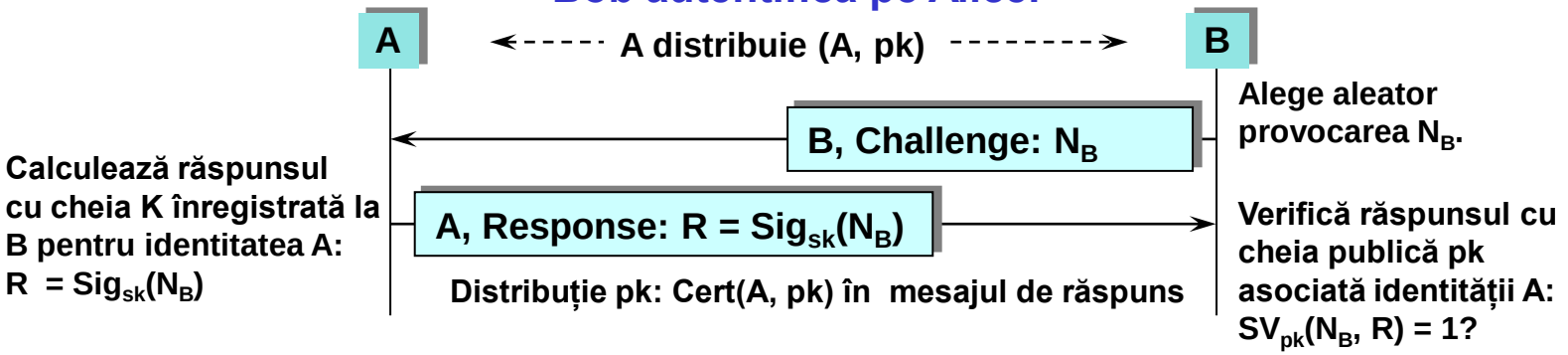


- Protocol bazat pe criptografie simetrică (cheie secretă).
- Înregistrare: A și B asociază identității A cheia secretă K (distribuție K).
- Autentificare: A demonstrează cunoașterea cheii calculând $\mathrm{R} = \mathrm{MAC}_\mathrm{K}(\mathrm{N})$.
- *Provocarea trebuie să fie un șir ales (pseudo)aleator (impredictibil)*.

- *Parametri de securitate*: $\ell = |\mathrm{N}|, \mathrm{k} = |\mathrm{K}|, \mathrm{n} = |\mathrm{MAC}_\mathrm{K}(\mathrm{N})|$.
- Securitate: Cum poate fi atacat UA-MAC dacă N este predictibil? Care este prob. ca N să se repete dacă este uniform aleator? Care este prob. ca adversarul să ghicească R dacă N nu este predictibil și nu se repetă?
- Criptografie simetrică: avantaje, dezavantaje?

Autentif. unilaterală cu semnătură

UA-SIG: Autentificare unilaterală cu semnătură.
Bob autentifică pe Alice.



- Protocol bazat pe criptografie asimetrică (cheie publică).
- Înregistrare: B obține cheia publică de semnătură a lui A (Cert(A, pk)).
- Pretendentul demonstrează cunoașterea cheii calculând $\mathrm{R} = \mathrm{Sig}_{\mathrm{sk}}(\mathrm{N})$.
- *Provocarea trebuie să fie un șir ales (pseudo)aleator (impredictibil).*

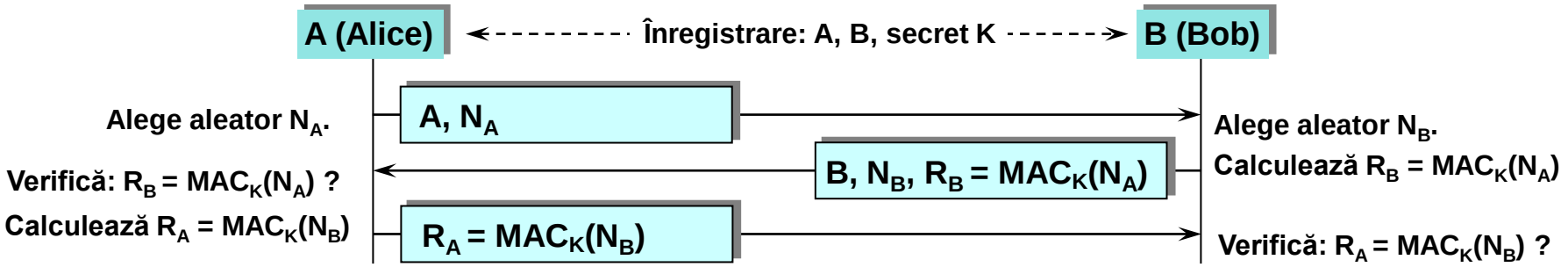
- *Parametri de securitate*: $\ell = |\mathrm{N}|$ și parametrii schemei de semnătură.
- Securitate: Cum poate fi atacat UA-SIG dacă N este predictibil? Care este prob. ca N să se repete dacă este uniform aleator? Cum poate adversarul să afle $\mathrm{R} = \mathrm{SigRSA}_{\mathrm{sk}}(\mathrm{N})$, pentru N nepredictibil, nerepetabil?
- Criptografie asimetrică: avantaje, dezavantaje?

Către autentificarea mutuală

Cum putem extinde UA-MAC pentru autentificare mutuală?

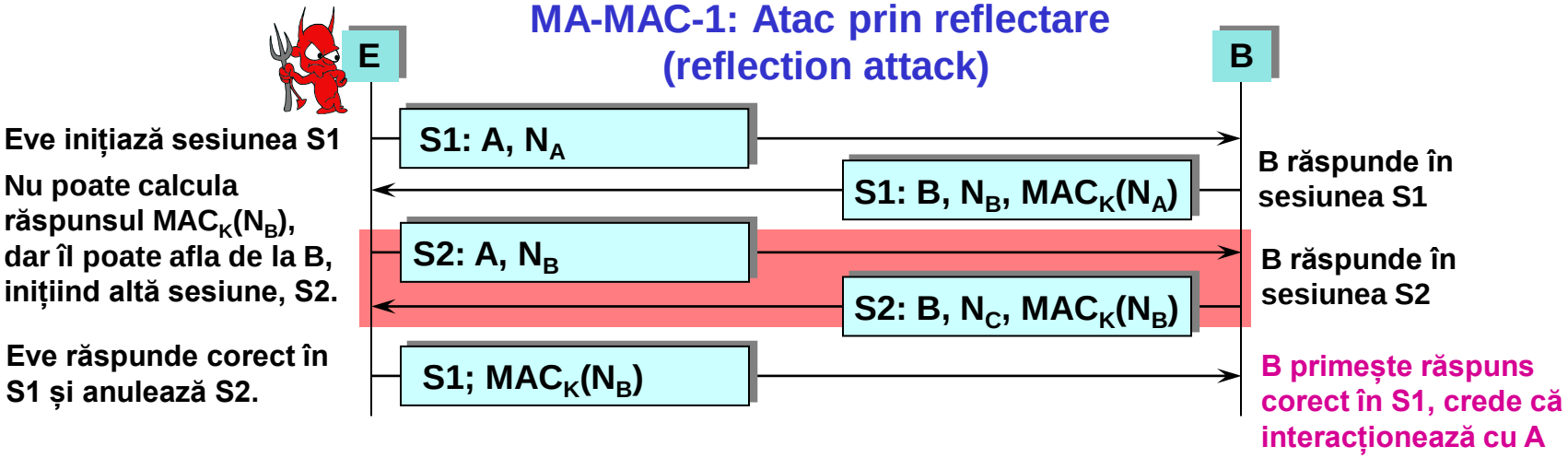
- Metoda evidentă: executăm UA-MAC de două ori.
- Înregistrare: A și B asociază o cheie K identitățile lor (sau câte o cheie diferită pentru fiecare identitate).
- Autentificare cu 4 mesaje: A autentifică B, apoi B autentifică A.
- Autentificare cu 3 mesaje: B trimite răspunsul împreună cu provocarea.

MA-MAC-1: Autentificare mutuală cu PSK și MAC



Este acest protocol sigur?

MA-MAC-1: Atac prin reflectare

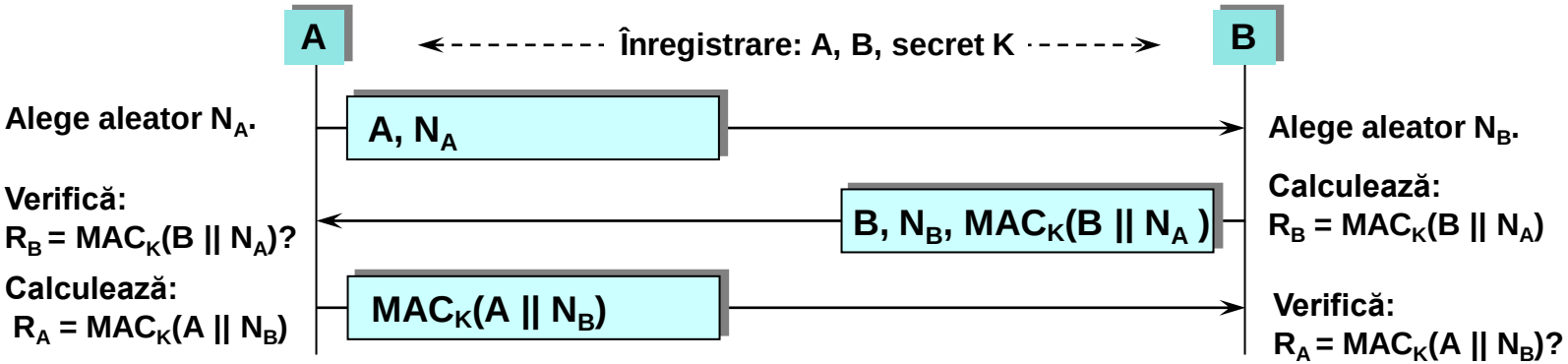


Cum putem să evităm atacurile prin reflectare?

- Vulnerabilitate în MA-MAC-1 care permite atacul prin reflectare: răspunsul este calculat la fel de ambii participanți (simetrie).
- Cum atașăm un răspuns de un singur participant? Mai multe soluții:
  - Introducem identitatea în calculul răspunsului.
  - Folosim chei diferite pentru fiecare participant (K_A, K_B).
  - Combinații ale metodelor precedente, etc.

Altă încercare: MA-MAC-2

MA-MAC-2: Autentificare mutuală cu PSK și MAC.
Răspunsul depinde de identitate: evită un atac prin reflectare

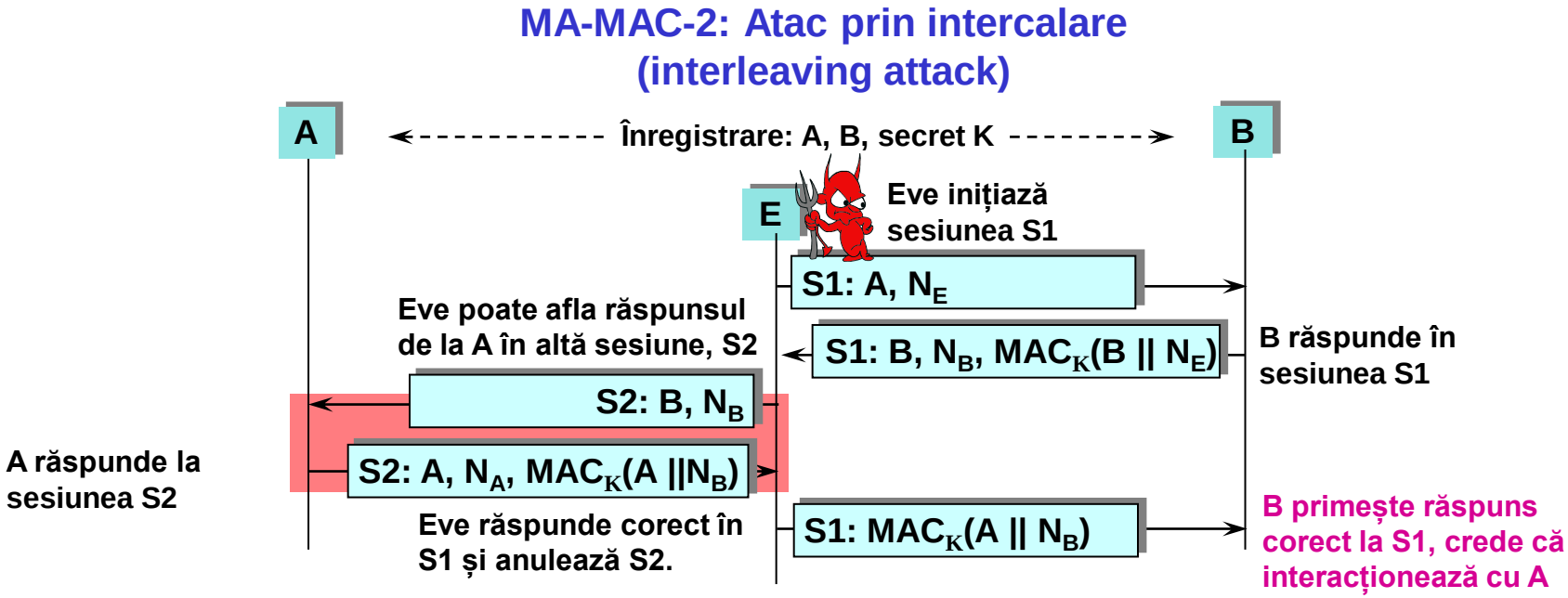


- De ce MA-MAC-2 nu este vulnerabil la atacuri prin reflectare?
  MA-MAC-2 în atacul prin reflectare: Eve poate obține prin reflectare $\mathrm{MAC_K(B \| N_B)}$, dar răspunsul corect este acum $\mathrm{MAC_K(A \| N_B)}$.

.

Este acest protocol sigur?

MA-MAC-2: Atac prin intercalare



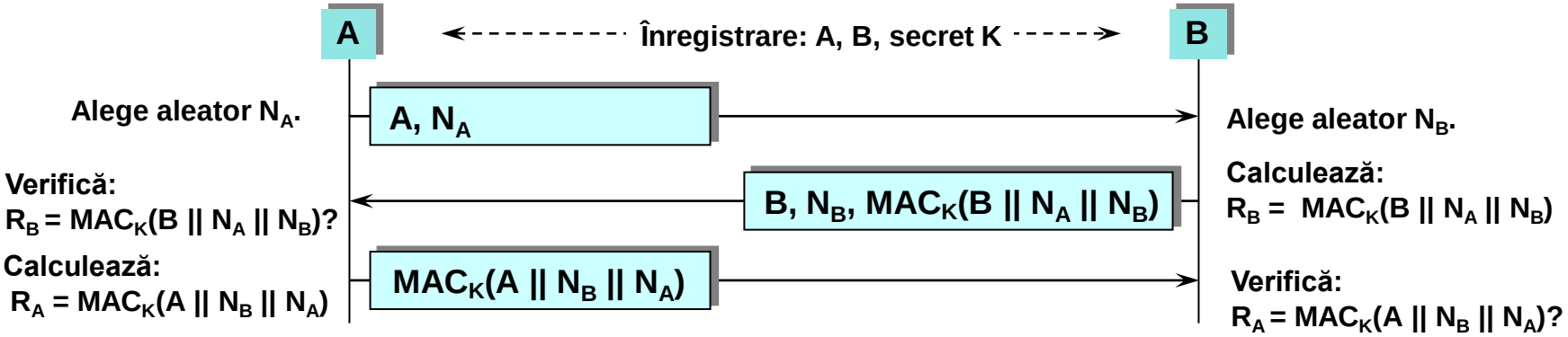
Cum putem să evităm atacurile prin intercalare (interleaving)?

- Vulnerabilitate în MA-MAC-2 care permite atacul prin intercalare: răspunsul dintr-o sesiune poate fi folosit în altă sesiune.
- Cum atașăm un răspuns unei singure sesiuni?

  Răspunsul trebuie să depindă de ambele provocări (evită și atacuri CMA asupra MAC). Mai robust: Răspunsul depinde și de identități.

MA-MAC: O variantă corectă

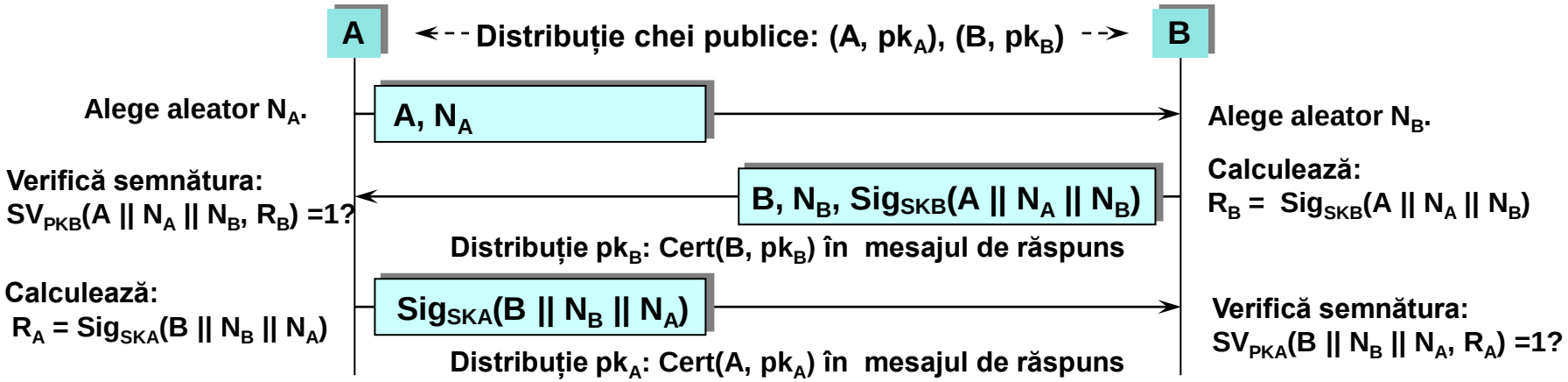
MA-MAC: Autentificare mutuală cu PSK și MAC (corect)



- De ce MA-MAC nu este vulnerabil la atacuri prin intercalare?
  MA-MAC în atacul prin intercalare: Eve obține $\mathrm{MAC_K(A \| N_B \| N_A)}$, dar răspunsul așteptat de B este acum $\mathrm{MAC_K(A \| N_B \| N_E)}$.
- *Parametri de securitate*: $\ell = |\mathrm{N}|, \mathrm{k} = |\mathrm{K}|, \mathrm{n} = |\mathrm{MAC_K(N)}|$.
- Pentru parametri de securitate suficient de mari, dacă $\mathrm{N_A}$ și $\mathrm{N_B}$ sunt alese uniform aleator și MAC este o funcție pseudo-aleatoare, un răspuns nu poate fi determinat de adversar și nici nu poate să apară în altă sesiune, decât cu probabilitate neglijabilă.
- De asemenea, răspunsurile sunt calculate cu intrări diferite (asimetrie).

Autentificare mutuală cu semnătură

MA-SIG: Autentificare mutuală cu semnătură



- Extensie a protocolului UA-SIG pentru autentificare mutuală.
- Variantă bazată pe criptografie asimetrică (cheie publică) a protocolului de autentificare mutuală MA-MAC. Analiză similară.
- Asimetrie inerentă datorată utilizării semnăturii în răspuns (chei private).
- MA-MAC sau MA-SIG? Comparație: Care variantă este mai potrivită (în diverse situații)? Avantaje, dezavantaje (performanțe, scalabilitate)?

Protocoale criptografice

Protocoale pentru stabilirea cheilor secrete

Managementul cheilor

Ciclul de viață al cheilor criptografice

- Cheile au o durată de viață limitată (număr limitat de operații):
  - Pentru a evita atacuri asupra algoritmilor criptografici.
  - Pentru a limita consecințele compromiterii lor.
- Operații implicate în ciclul de viață al cheilor: generare, distribuire, instalare, stocare, utilizare, distrugere; eventual arhivare și înlocuire.

Principiul separării cheilor

- Compromiterea securității unei aplicații criptografice nu trebuie să afecteze securitatea altor aplicații.
- O cheie criptografică trebuie să aibă un scop bine definit și nu trebuie utilizată niciodată în alt scop. $\Rightarrow$ Chei diferite pentru: autentificarea datelor, confidențialitatea datelor, autentificarea entităților, etc.

Managementul cheilor criptografice

- Ansamblu de tehnici și proceduri care asigură stabilirea, utilizarea și întreținerea cheilor utilizate în criptografia simetrică și asimetrică.

Distribuția cheilor

Distribuția cheilor pentru criptografie asimetrică (cheie publică)

- Cerințe: Trebuie distribuită cheia publică a unui participant către orice partener de comunicație, asigurând autenticitatea cheii.
- Soluție generală, scalabilă: infrastructură pentru chei publice (certificate).

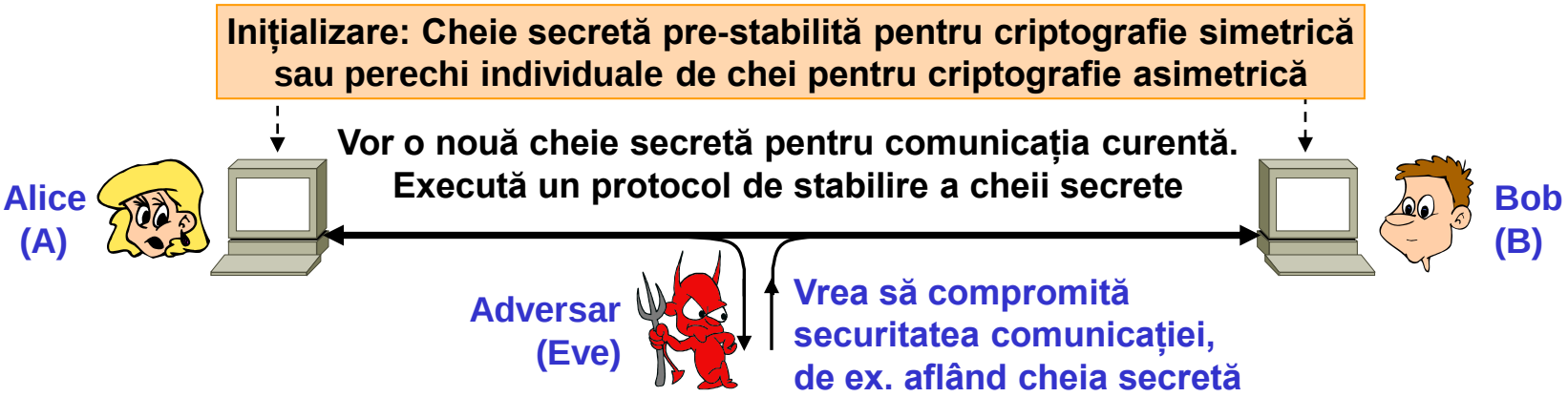
Distribuția cheilor pentru criptografie simetrică (cheie secretă)

- Cerințe: Trebuie distribuite chei secrete diferite pentru orice pereche de parteneri de comunicație, asigurând confidențialitatea și autenticitatea lor.
- Soluție generală, scalabilă: protocoale pentru stabilirea cheilor.

Stabilirea cheilor secrete (KE: Key Exchange / Key Establishment)

- Chei de sesiune (session key): Chei efemere, utilizate pentru a proteja o singură comunicație, stabilite la inițializarea comunicației.
- Protocoalele pentru stabilirea cheilor de sesiune asigură autenticitatea și confidențialitatea acestora folosind criptografie simetrică sau asimetrică și chei pe termen lung (long-lived keys) dedicate acestui scop.

Stabilirea cheilor secrete (KE)



Protocoale criptografice pentru stabilirea cheilor secrete (KE)

- Scenariu: Doi participanți, A și B, comunică printr-un canal controlat de adversar și execută un protocol criptografic de stabilire a cheilor secrete.
- Există și protocoale în care intervine un al 3-lea participant, de încredere (trusted 3rd party). Ne ocupăm de protocoalele cu 2 participanți.

- Adversarul: Capabilitățile adversarului sunt similare celor din cazul protocoalelor de autentificare. Obiectivele sunt însă diferite.
- Cum pot fi compromise aplicațiile care folosesc un astfel de protocol? Ce proprietăți de securitate trebuie să îndeplinească protocolul?

Securitatea protocoalelor KE (1)

Confidențialitatea și autenticitatea cheii

- Dacă A dorește să stabilească o cheie secretă cu B și primește cheia K (în urma executării protocolului), atunci *protocolul trebuie să îi ofere dovezi nefalsificabile că cheia K poate fi cunoscută doar de B*.
- Proprietate fundamentală care combină autentificarea entităților, confidențialitatea cheii și autenticitatea cheii. $\Rightarrow$ *Protocoale de stabilire a cheii cu autentificare (Authenticated Key Exchange (AKE))*.
- Autentificare mutuală sau (pentru aplicații care permit unui participant să rămână anonim) autentificare unilaterală.

Idee de construcție: Combinăm un protocol de autentificare și o metodă de transport confidențial al cheilor de sesiune (integrare).

Protocoale care folosesc criptografie simetrică și chei secrete stabilite în prealabil (PSK).

Protocoale care folosesc criptografie asimetrică și infrastructură pentru chei publice.

Protocoale care combină criptografie simetrică și criptografie asimetrică.

Securitatea protocoalelor KE (2)

Confirmarea cheii

- Un protocol asigură confirmarea (explicită a) cheii dacă fiecare participant obține dovada că partenerul cunoaște efectiv cheia stabilită.
- Protocoalele fără confirmarea cheii oferă doar garanții că nici un alt participant nu poate cunoaște cheia stabilită.

Independența cheilor

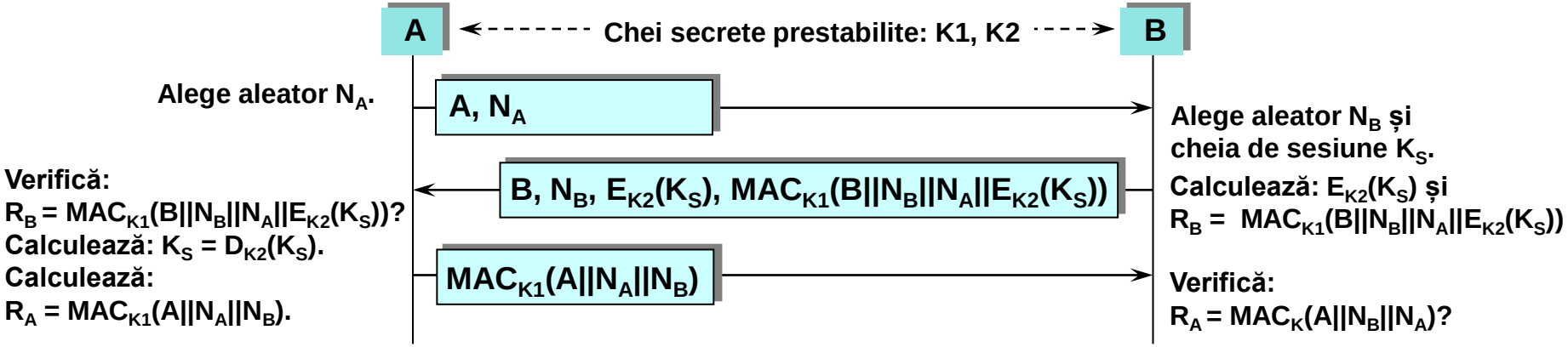
- Un protocol asigură independența cheilor dacă compromiterea cheii stabilite într-o sesiune nu afectează securitatea altor sesiuni.
- În acest scop, pentru fiecare sesiune, protocolul trebuie să folosească în calculul cheii valori alese aleator și indepedent.

Protejarea sesiunilor anterioare (Perfect Forward Secrecy (PFS))

- Un protocol oferă PFS dacă în urma compromiterii cheilor pe temen lung sau a oricăror chei de sesiune nu sunt compromise sesiunile anterioare. ("All previous traffic is safely locked in the past").

KE cu chei secrete pre-stabilite (1)

KE-PSK-MAC-1: Exemplu de protocol (naiv) cu chei secrete prestabilite



- KE-PSK-MAC-1 combină protocolul de autentificare mutuală MA-MAC și transportul cheii de sesiune cu criptare autentificată.
- Asigură: confidențialitatea și autenticitatea cheii, indepedența cheilor.

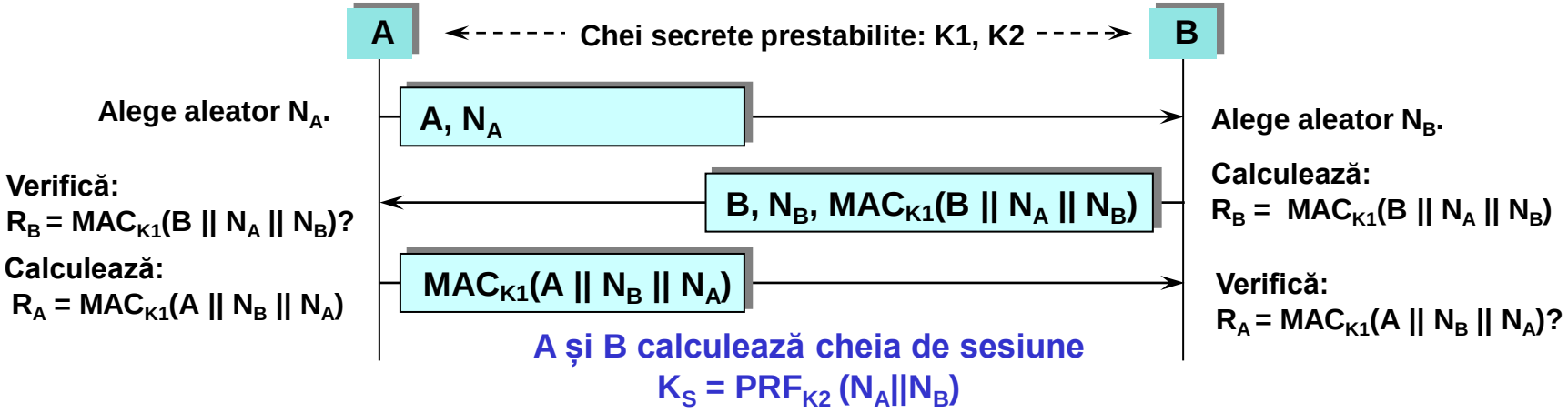
  Ce garanții oferă participanților legitimi?

  Cum previne atacurile prin intercalare?
- Nu asigură: confirmarea cheii și PFS.

  Ce se întâmplă dacă cheile pe termen lung sunt compromise?

KE cu chei secrete pre-stabilite (2)

KE-PSK-MAC-2: Protocol cu chei secrete prestabilite și autentificare cu MAC



- KE-PSK-MAC-2 extinde protocolul de autentificarea mutuală MA-MAC cu o metodă de generare a cheii de sesiune folosind PRF (ex: MAC).
  Mai eficient decât KE-PSK-MAC-1, evită transportul criptat al cheii.
- Asigură: confidențialitatea și autenticitatea cheii, indepedența cheilor.
  Ce garanții oferă privind confidențialitatea și autenticitatea cheii?
- Nu asigură: confirmarea cheii și PFS.
  Ce se întâmplă dacă cheile pe termen lung sunt compromise?

Criptografie cu logaritmi discreți (1)

Parametrii unui algoritm criptografic bazat pe logaritmi discreți

- Considerăm un grup ciclic G de ordin q și $g \in G$ un generator al grupului.
- Prin definiție (grup ciclic G, ordin q, generator g): $G = \{ g^0, g^1, g^2, ..., g^{q-1} \}$.

Funcția exponențială discretă

$DExp_{G,g} : [0, q-1] \rightarrow G$,

$DExp_{G,g}(x) = g^x$, este bijectivă.

Funcția logaritm discretă

$DLog_{G,g} : G \rightarrow [0, q-1]$,

$DLog_{G,g}(g^x) = x$.

Problema logaritmului discret (Discrete Logarithm (DL))

- Fiind dat $y \in G$, găsiți numărul întreg $x \in [0, q-1]$ astfel încât $g^x = y$.
- În criptografie, vom folosi grupuri în care problema DL este considerată necalculabilă: algoritmii DL au complexitate exponențială sau sub-exponențială (exemple în secțiunea despre permutări cu sens unic).
- Pentru a obține scheme mai eficiente, preferăm grupuri în care algoritmii DL au complexitate exponențială: $O(2^{k/2})$, unde $k = \log_2(q)$.

Criptografie cu logaritmi discreți (2)

Problema Diffie-Hellman (Computational DH (CDH))

- Fiind date elementele $g^x, g^y \in G$, găsiți elementul $g^{xy} \in G$.
- Dacă putem rezolva eficient problema DL, atunci putem rezolva la fel de eficient problema CDH. Reciproca nu este în general adevărată.

Problema Diffie-Hellman decizională (Decisional DH (DDH))

- Fiind date elementele $g^x, g^y, g^z \in G$, decideți dacă $g^{xy} = g^z$.
- Dacă putem rezolva eficient problema CDH, atunci putem rezolva la fel de eficient problema DDH. Reciproca nu este în general adevărată.

- Ne interesează în special grupuri în care problema DL are complexitate exponențială și problema DDH este (considerată) necalculabilă.
- Principalul atac constă în rezolvarea problemei DL, cu effort $O(2^{k/2})$, unde $k = \log_2(q)$. Principalul parametru de securitate este deci k.
- Exemple: Subgrup de ordin prim al grupului multiplicativ Z^*_p, cu p prim. Subgrup de ordin prim al punctelor de pe o curbă eliptică discretă.

Protocolul Diffie-Hellman (DH)

- Parametri publici: Grup ciclic G de ordin prim q în care problema DDH este nerezolvabilă. Generator $g \in G$.
- Protocolul de stabilire a cheii, cu participanții A și B:

A alege $x \in [1, q-1]$, uniform aleator, calculează $X = g^x$, și trimite X lui B.

B alege $y \in [1, q-1]$, uniform aleator, calculează $Y = g^y$, și trimite Y lui A.

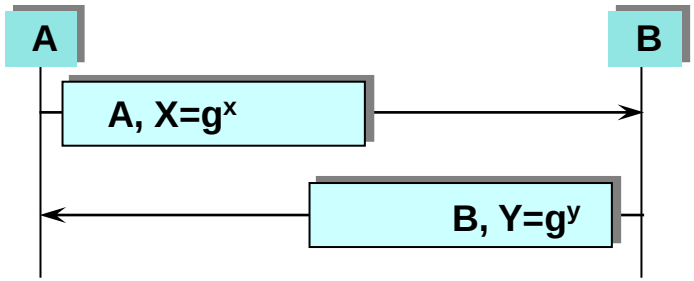
- Calculul secretului:

A calculează $S = Y^x = g^{xy}$.

B calculează $S = X^y = g^{xy}$.

Alege aleator $x \in [1, q-1]$
Calculează $X = g^x$.

Calculează secretul:
$S = Y^x = (g^y)^x = g^{xy}$



Alege aleator $y \in [1, q-1]$
Calculează $Y = g^y$.

Calculează secretul:
$S = X^y = (g^x)^y = g^{xy}$

Cheile de sesiune sunt calculate aplicând PRF secretului S

Securitatea protocolului DH

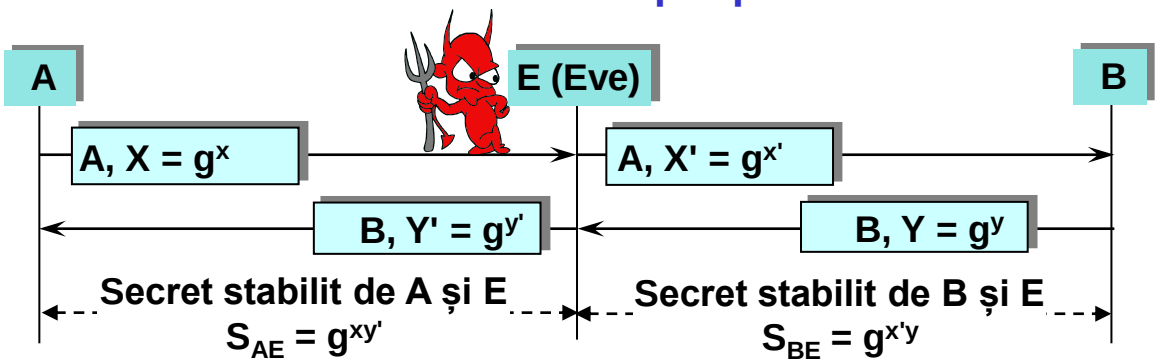
Ipoteza DDH

- Dacă $x, y \in [0, q\text{-}1]$ sunt alese uniform aleator, probabilitatea ca un adversar cu resurse limitate care știe g^x, g^y să distingă g^{xy} de un element din G ales uniform aleator este neglijabilă, dacă q este suficient de mare.
- Implică ipoteze similare pentru problemele DL și CDH.

- Pentru un grup în care ipoteza DDH este validă, protocolul DH asigură confidențialitatea cheii stabilite. Parametrul de securitate este $k = \log_2(q)$.

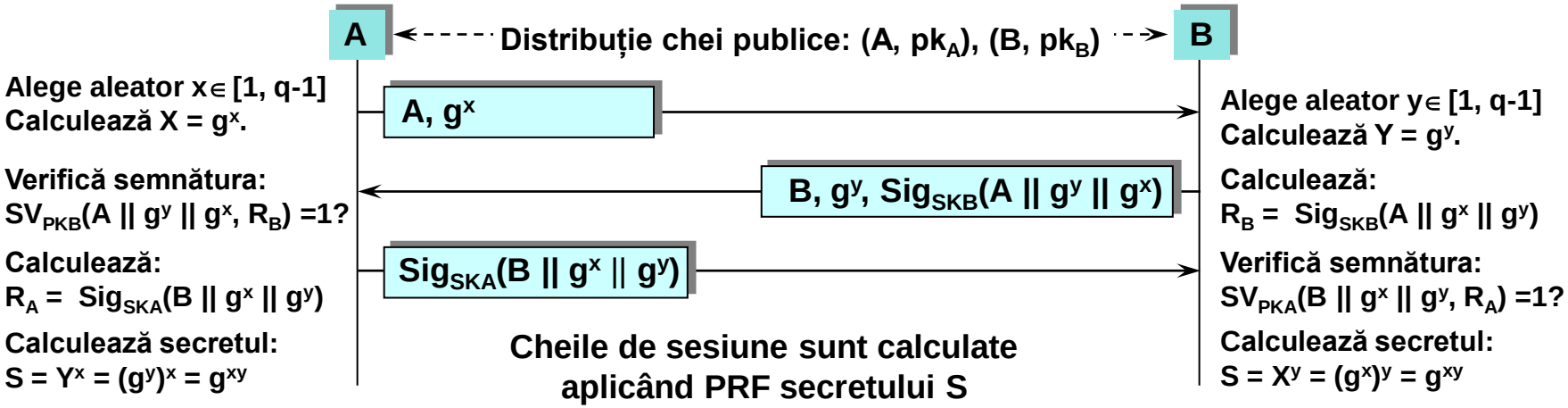
- Vulnerabilitate fatală: Protocolul DH nu asigură autentificarea cheii.

Atac man-in-the-middle asupra protocolului DH



KE cu DH și semnături

Protocol KE-DH-SIG: Protocol cu DH și autentificare mutuală prin semnături.



- KE-DH-SIG extinde protocolul de autentificarea mutuală MA-SIG cu o metodă de generare a cheii de sesiune folosind protocolul DH.
- Asigură: confidențialitatea și autenticitatea cheii, indepedența cheilor.
  Ce garanții oferă privind confidențialitatea și autenticitatea cheii?
- Asigură PFS (dacă x și y sunt alese aleator și distruse după utilizare).
  Ce se întâmplă dacă cheile pe termen lung sunt compromise?
- Nu asigură confirmarea cheii.