Atak na protokół Needhama-Schroedera

- Seminarium

  Protokoły komunikacyjne

- Autor

  Piotr Witusowski

14-12-2004

Wstęp

- O protokole
- Klucze
- Kroki protokołu
- Przebieg ataku
- Poprawka Lowe'a
- Uogólnienie na większe systemy
- Modelowanie protokołu i ataku
- Automatyczna weryfikacja

Wprowadzenie

Protokół Needhama-Schroedera:

- Powstał w 1978 roku.
- Oparty na szyfrowaniu przy pomocy kluczy: publicznego i prywatnego.
- Jego celem jest ustanowienie wzajemnej autoryzacji między ***inicjatorem A*** a ***odbiorcą B,*** po której może nastąpić sesja, służąca ***A*** i ***B*** do wymiany wiadomości.

Klucze

- Każdy agent ***H*** posiada swój własny ***klucz publiczny*** oznaczany jako ***PK{H},*** oraz ***klucz prywatny SK{H}***
- ***Klucz prywatny SK{H}:***
  - pownien być znany tylko ***H***;
  - pozwala na podpisywanie wiadomości;
  - dekoduje wiadomości zaszyfrowane przy pomocy klucza ***PK{H}***.

Klucze

- ***Klucz publiczny PK{H}:***
  - każdy agent może go pobrać z serwera kluczy;
  - dekoduje wiadomości zaszyfrowanych przy pomocy ***SK{H}***;
  - umożliwia szyfrowanie wiadomości.

Szyfrowanie

- Każdy agent może zaszyfrować wiadomość ***x*** używając klucza publicznego ***H*** aby uzyskać zakodowaną wiadomość ***{x}PK{H}***;
- Tylko agent znający klucz prywatny ***H*** może zdekodować wiadomość ***{x}PK{H}*** i otrzymać ***x*** – zapewnia to bezpieczeństwo ***x***;
- Każdy agent ***H*** może podpisać wiadomość kodując ją swoim kluczem prywatnym do ***{x}SK{H}*** aby zapewnić integralność ***x.***
- Każdy agent może rozszyfrować ***{x}SK{H}*** używając klucza publicznego ***H.***

Założenia

Kilka założeń pozwalających oddzielić sprawdzanie poprawności protokołu od weryfikacji systemów szyfrujących:

- jedyną metodą rozkodowania zaszyfrowanej wiadomości jest znajomość odpowiedniego klucza;
- zaszyfrowana wiadomość nie zdradza przy użyciu którego klucza została zakodowana;
- istnieje możliwość sprawdzenia przez algorytm rozszyfrowujący, czy wiadomość została zakodowana przy użyciu odpowiedniego klucza.

Wersja pełna protokołu

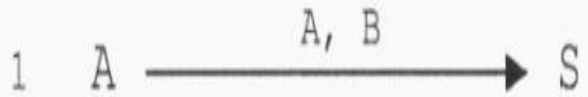
W protokół zaangażowanych jest trzech agentów:

- Inicjator ***A*** – chce ustanowić sesję z ***B***
- Odbiorca ***B*** – to z nim ***A*** chce ustanowić sesję
- Serwer kluczy ***S*** – przechowuje klucze publiczne wszystkich agentów

- Zakładamy że agenci znają ***klucz publiczny PK{S}***.

Atak na protokół Needhama-Schroedera

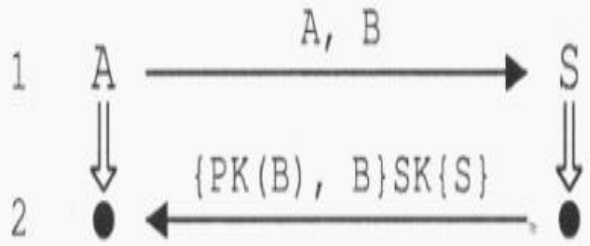
Krok 1



- ***A*** wysyła do ***S*** komunikat o treści ***A, B*** – czyli ***A*** prosi ***S*** o ***PK{B}*** – ***klucz publiczny B***.

Atak na protokół Needhama-Schroedera

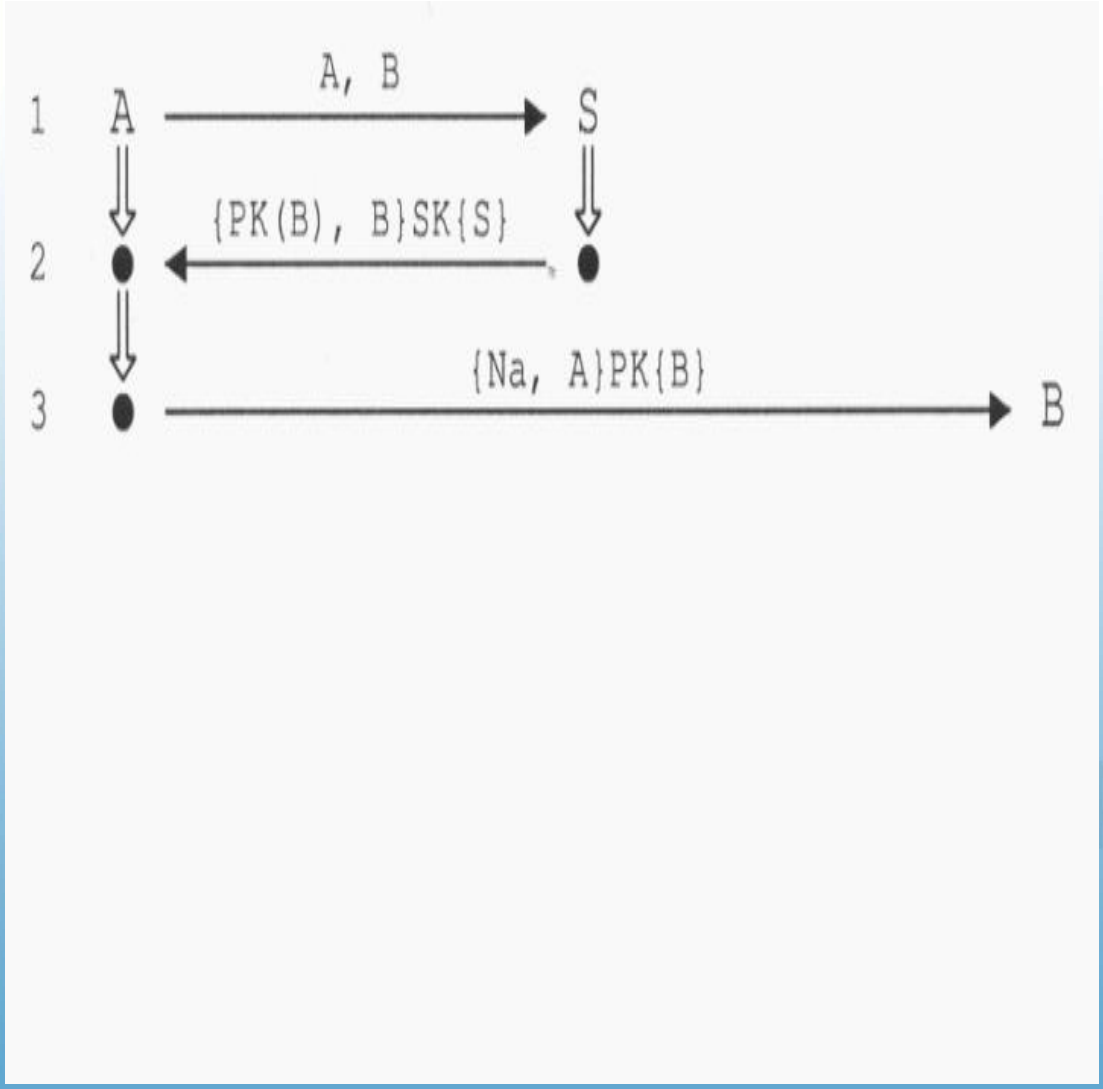
Krok 2



- ***S*** odsyła ***PK{B}*** dodając identyfikator ***B*** zaszyfrowane swoim kluczem ***SK{S}*** (prywatnym) w celu zapewnienia integralności danych.
- Od tej chwili ***A*** zna ***PK(B)***.

Atak na protokół Needhama-Schroedera

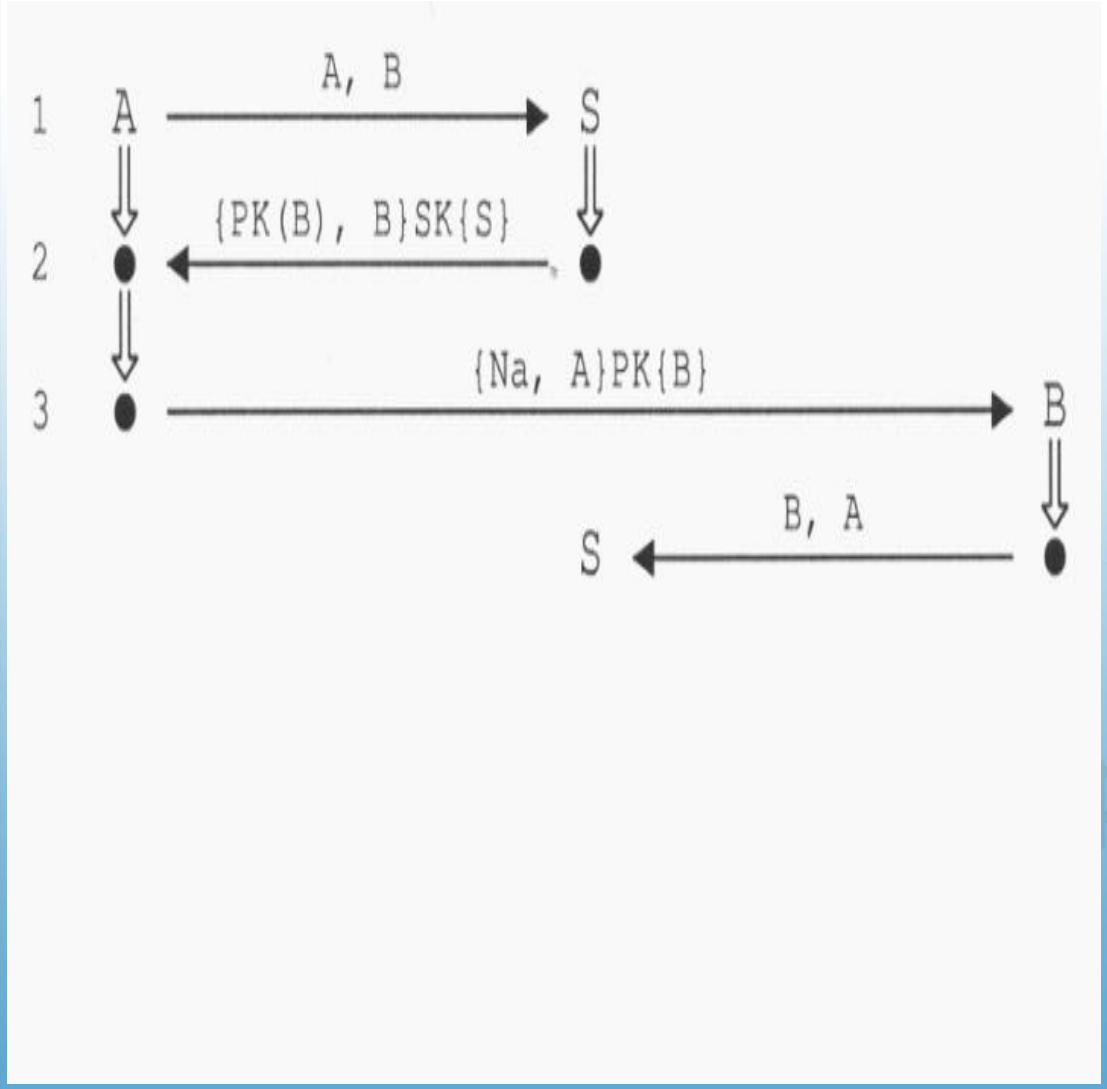
Krok 3



- ***A*** generuje znacznik ***Na*** (nie da się go zgadnąć).
- Wysyła znacznik ***Na*** i własny identyfikator zaszyfrowane ***PK{B}***.
- Wiadomość może rozszyfrować tylko agent znający klucz ***SK{B}***.

Atak na protokół Needhama-Schroedera

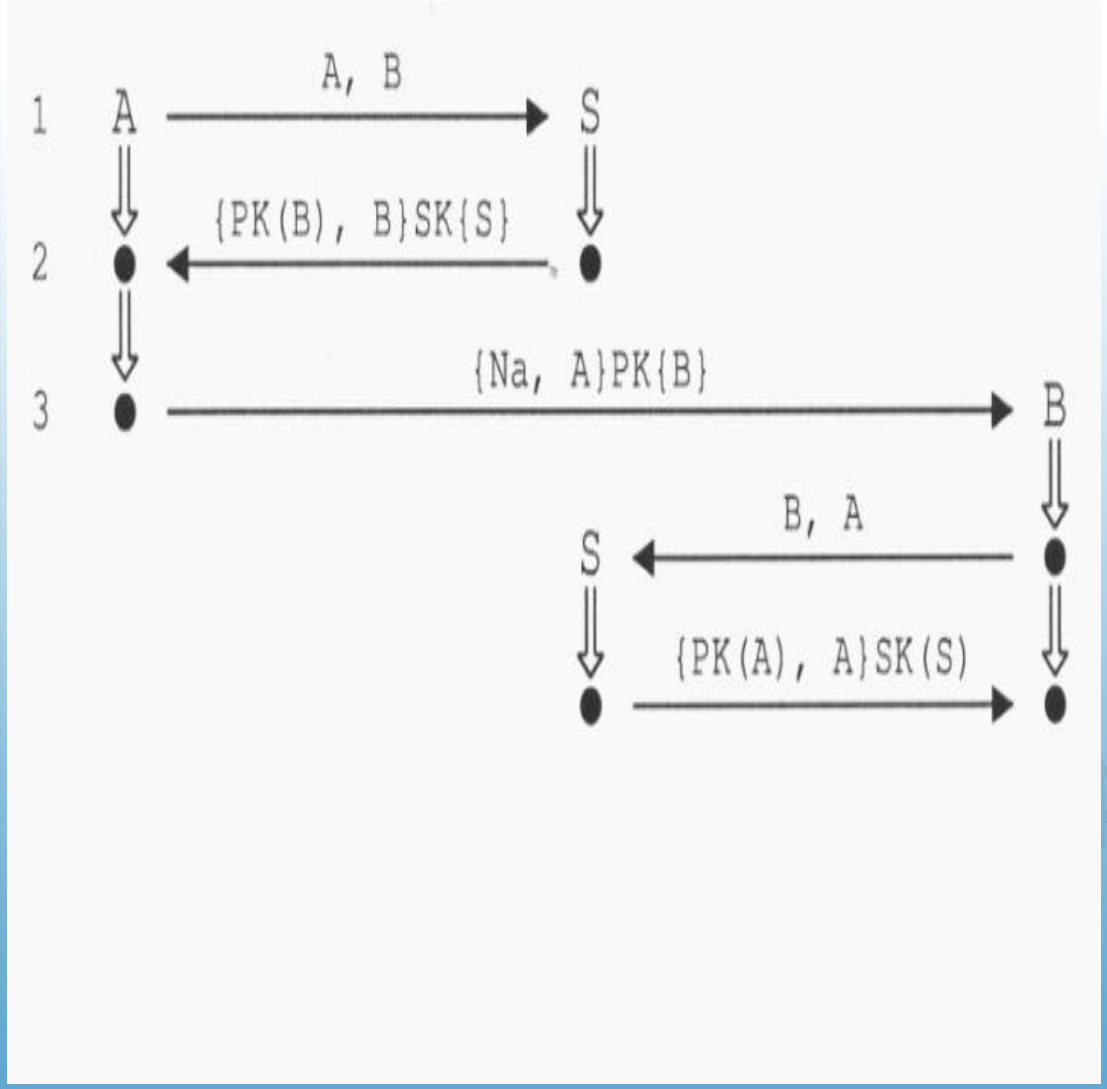
Krok 4



- ***B*** prosi ***S*** o ***PK{A}*** (podobnie ***A*** prosił o klucz ***PK{B}***).

Atak na protokół Needhama-Schroedera

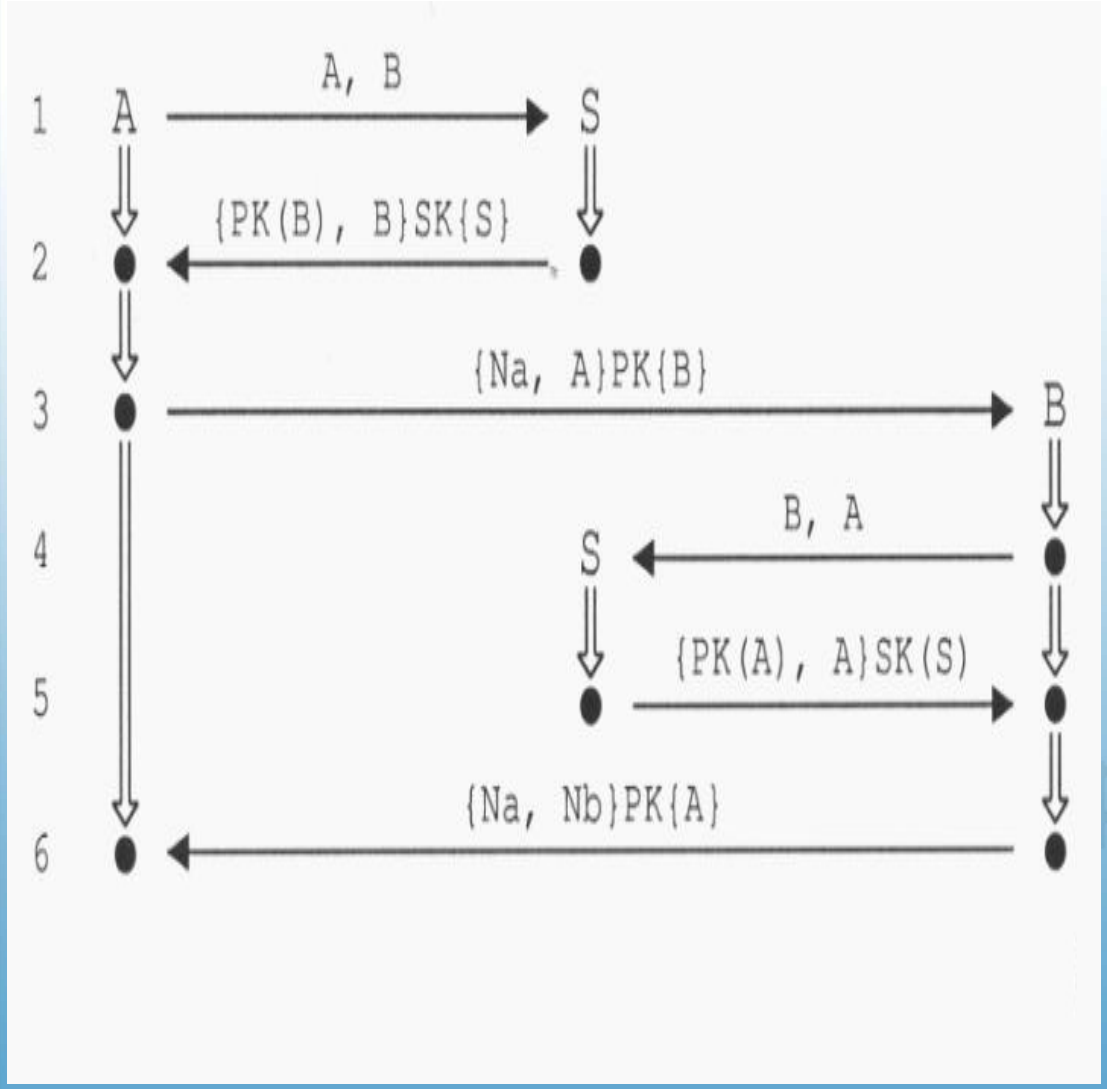
Krok 5



- ***S*** odsyła klucz publiczny ***PK{A}*** i identyfikator ***A*** podpisane swoim kluczem prywatnym.

Atak na protokół Needhama-Schroedera

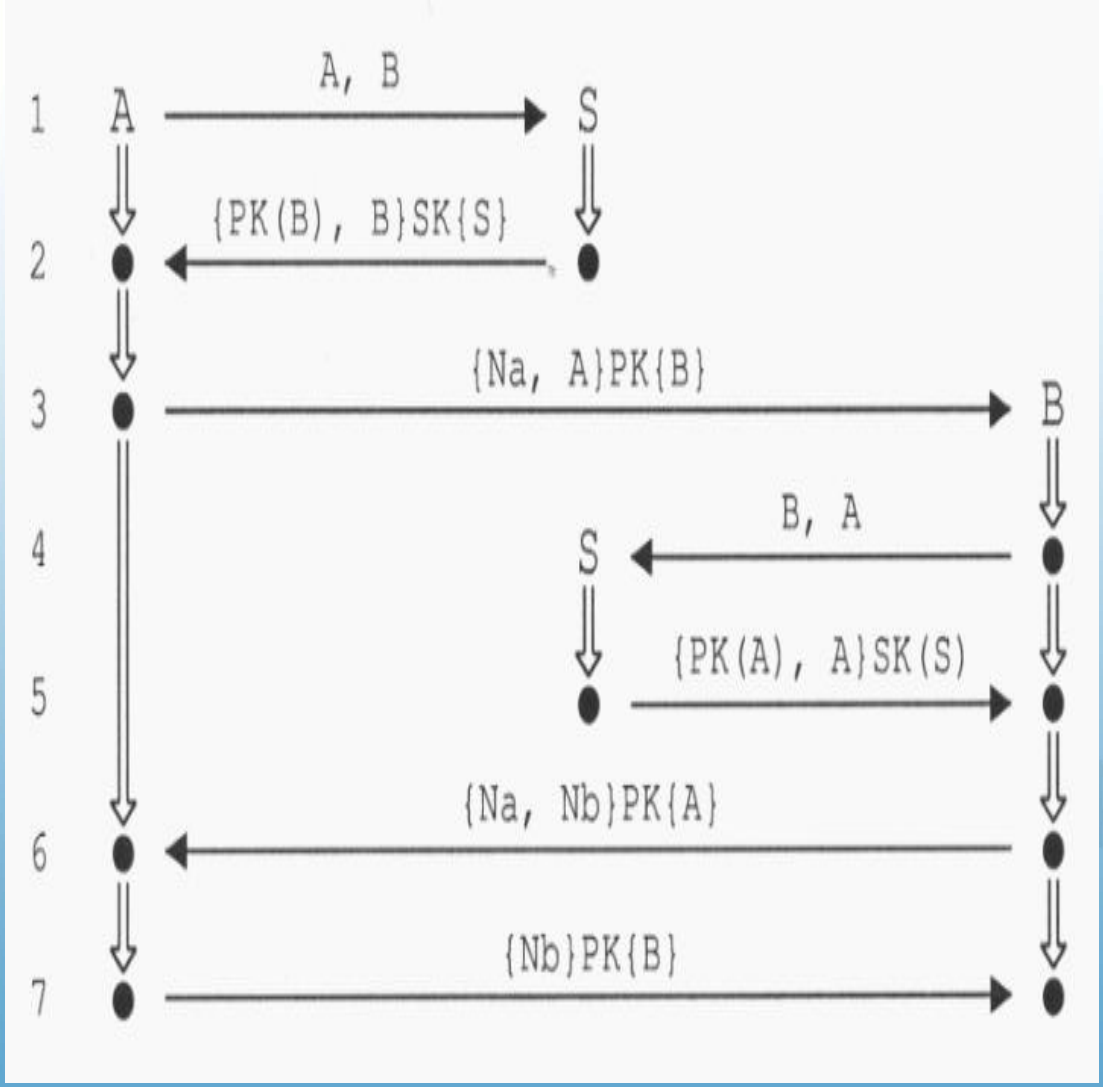
Krok 6



- ***B*** generuje znacznik ***Nb***.
- Wysyła znaczniki ***Na*** i ***Nb*** zaszyfrowane otrzymanym przed chwilą kluczem publicznym ***A***.

Atak na protokół Needhama-Schroedera

Krok 7

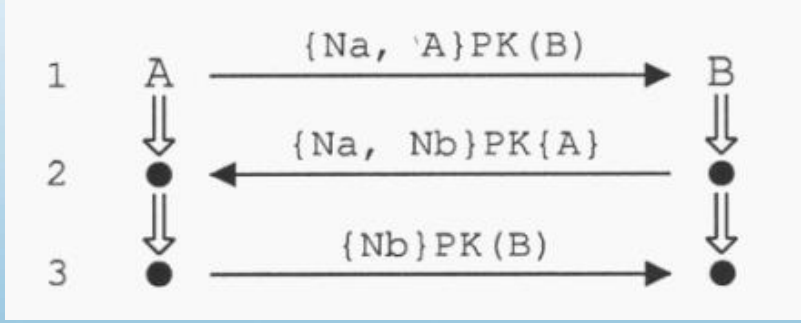


- ***A*** rozkodowuje wiadomość
- Sprawdza czy otrzymał ***Na***.
- Jeżeli tak to odsyła znacznik ***Nb*** zaszyfrowany kluczem publicznym ***B***.
- ***B*** sprawdza czy otrzymał ***Nb***.
- **Sesja jest ustanowiona.**

Możemy trochę uprościć

- Załóżmy, że wszyscy znają swoje klucze publiczne (przed przebiegiem protokołu zapytali się ***S*** o wszystkie możliwe klucze)
- W takim wypadku możemy pozbyć się serwera kluczy z naszego protokołu
- Usuwamy kroki 1, 2 i 4, 5, które służyły pobieraniu kluczy publicznych od ***S***.

Wersja uproszczona



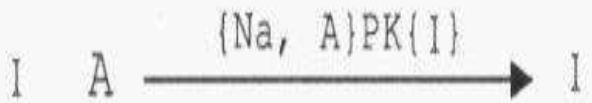
Intruz

- Reprezentuje wszystkich potencjalnych atakujących
- Jest normalnym użytkownikiem sieci
- Przestrzega założeń

ATAK NA PROTOKÓŁ

Atak na protokół Needhama-Schroedera

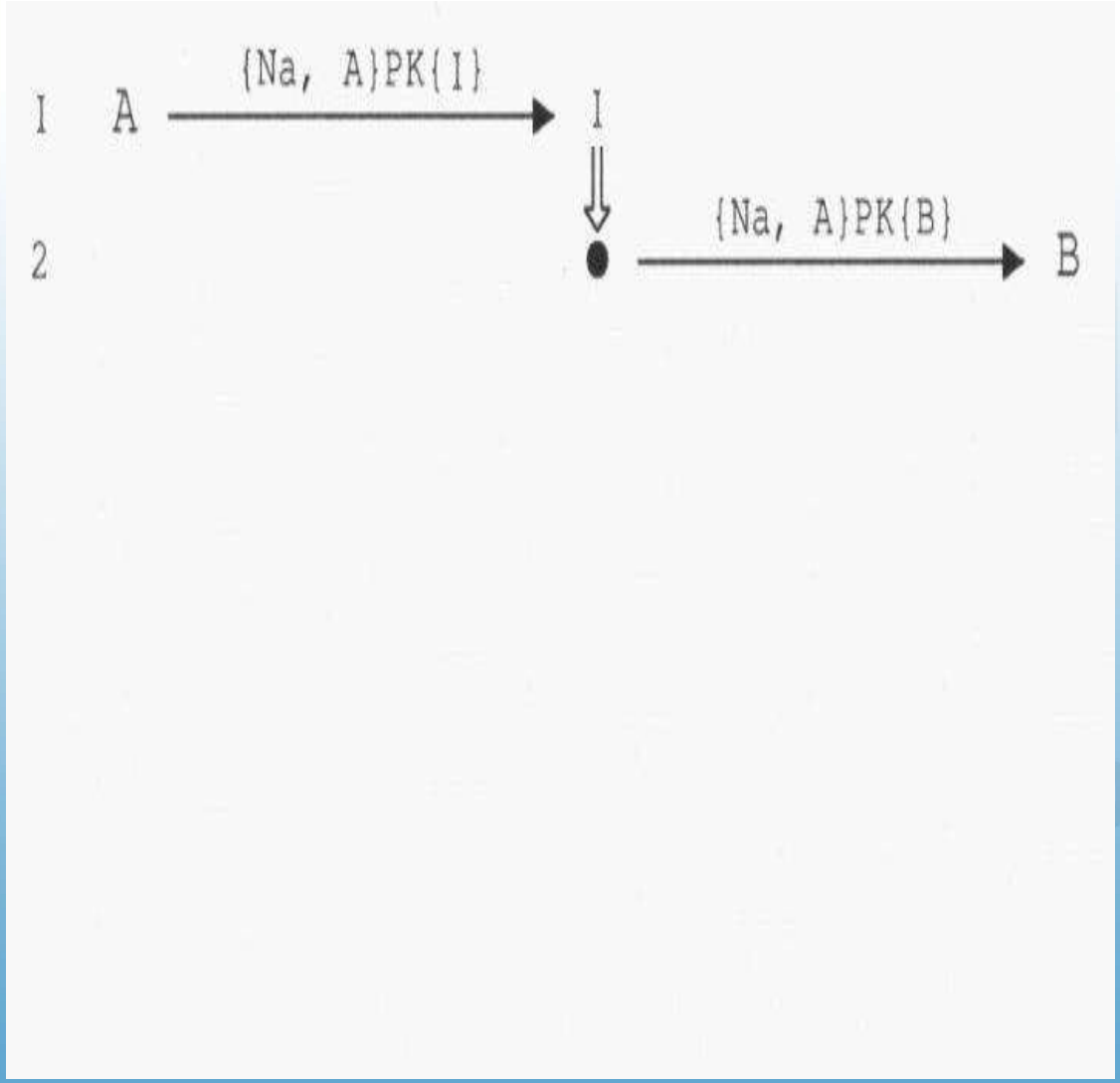
Krok 1



- ***A*** inicjuje sesję z ***I***.
- ***I*** poznaje ***Na***.

Atak na protokół Needhama-Schroedera

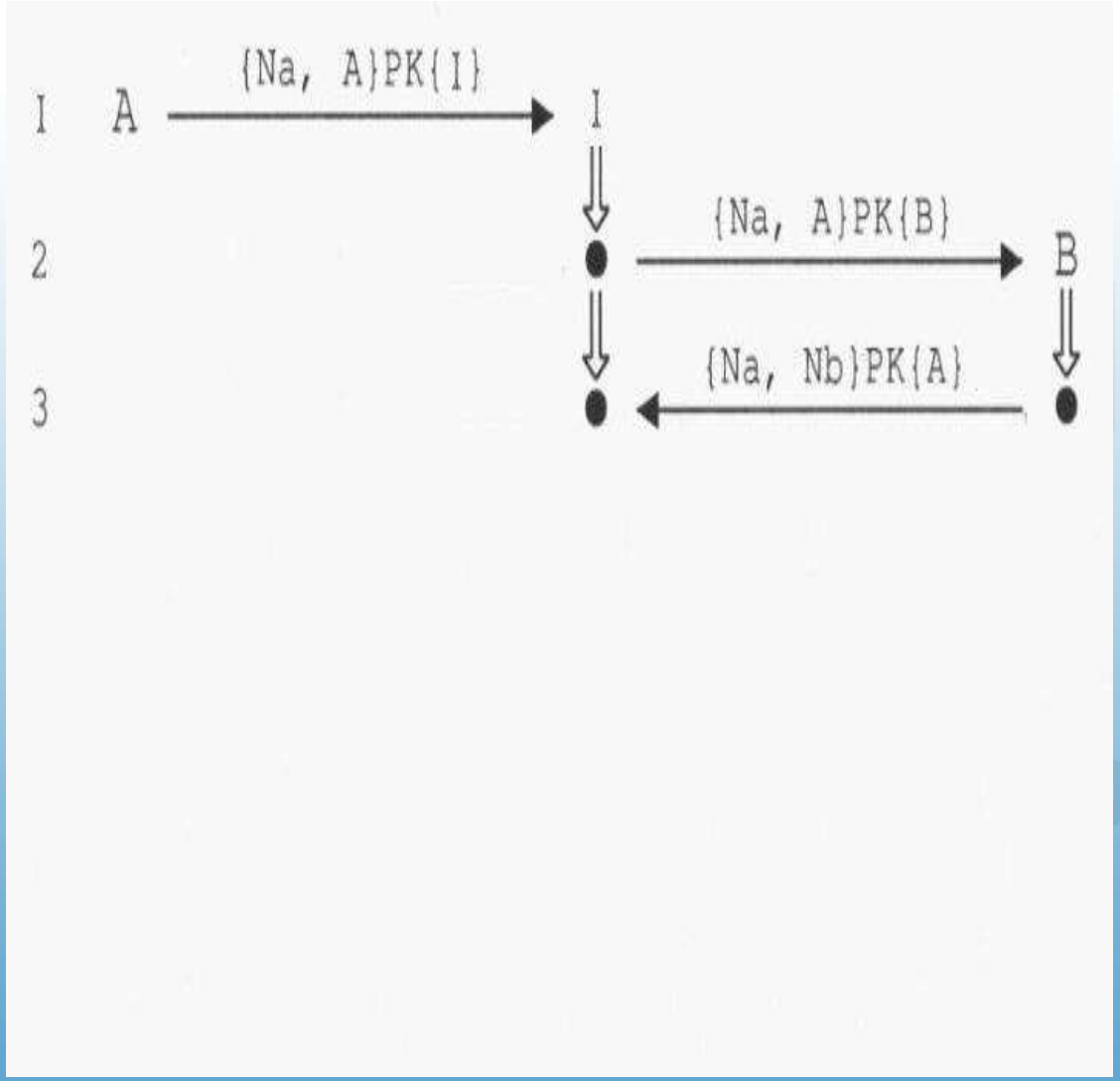
Krok 2



- ***I*** inicjuje sesję z ***B***, używając ***Na*** i ***A*** (identyfikatora) – podszywa się pod ***A***.
- ***B*** odbiera wiadomość, dekoduje...

Atak na protokół Needhama-Schroedera

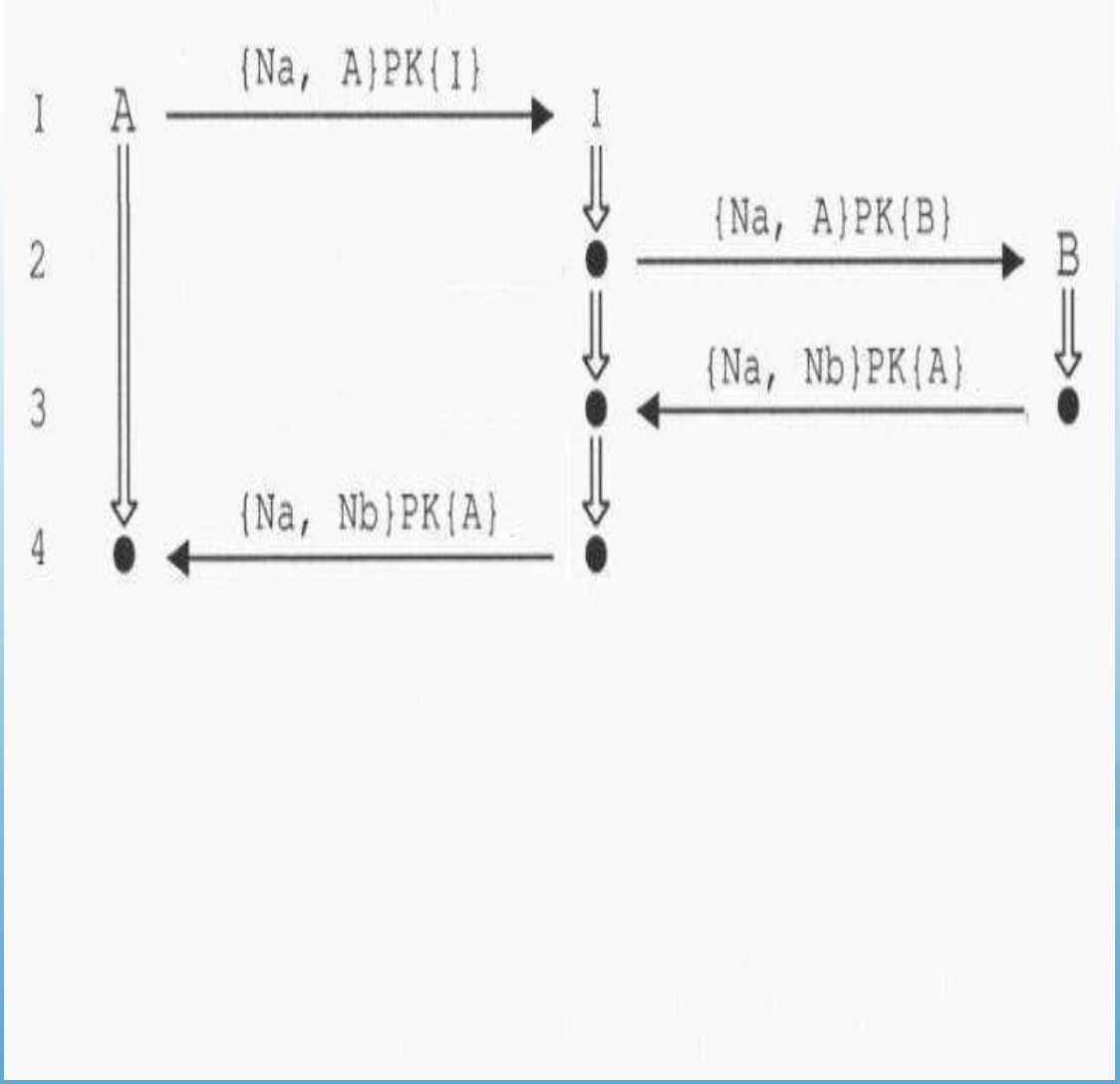
Krok 3



- ***B*** odbiera wiadomość, dekoduje...
- Generuje znacznik ***Nb*** i odsyła razem z ***Na*** zakodowane ***PK{A}***.
- ***I*** nie zna ***SK{A}*** więc nie może zdekodować wiadomości...

Atak na protokół Needhama-Schroedera

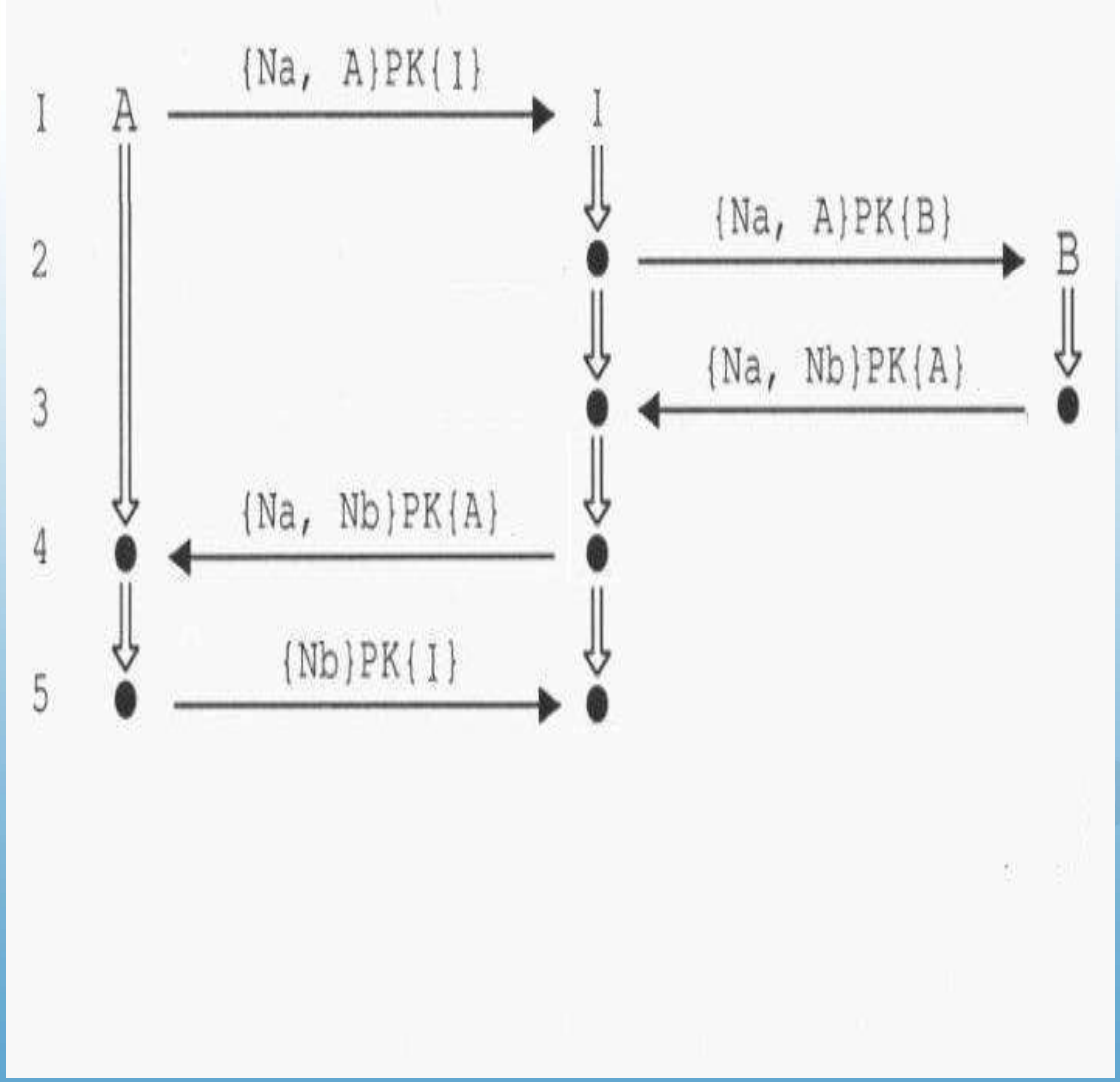
Krok 4



- Ale może przesłać wiadomość do ***A***, które zna ***SK{A}***.
- ***A*** rozszyfrowuje wiadomość.
- ***A*** sprawdza czy otrzymał ***Na***, zna ***Nb*** i ...

Atak na protokół Needhama-Schroedera

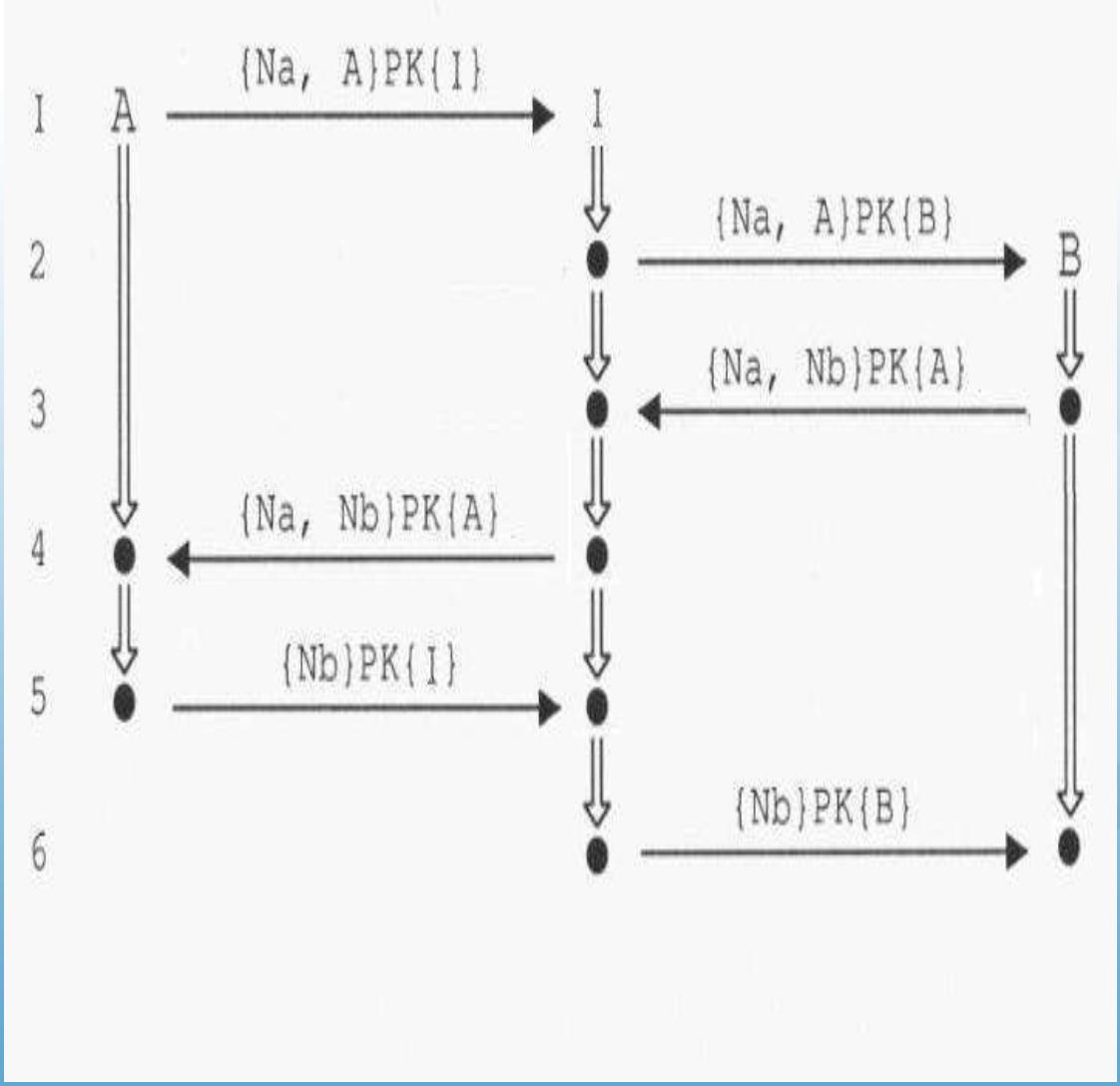
Krok 5



- ***A*** grzecznie wysyła ***Nb*** zakodowane kluczem publicznym ***PK{I}***.
- ***I*** dekoduje ***{Nb}PK{I}*** i poznaje ***Nb***.
- Skoro ***I*** zna już ***Nb***...

Atak na protokół Needhama-Schroedera

Krok 6



- To wysyła go ***B***

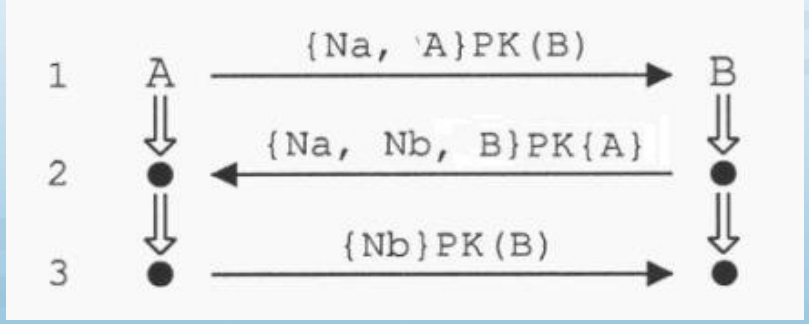
Skutki:

- ***A*** ma ustanowioną sesję z ***I***
- ***I*** ma ustanowioną sesję z ***B*** ale jako ***A***

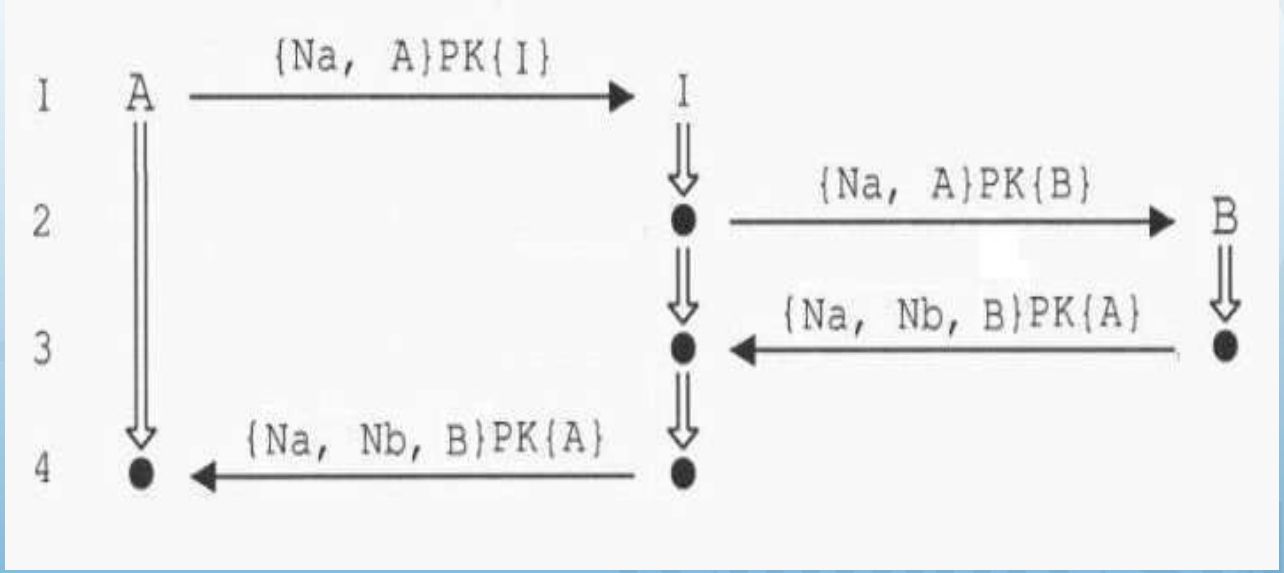
Poprawka Gavina Lowe'a

- Rok 1995 (17 lat od powstania protokołu)
- Bardzo prosta...

Poprawka Gavina Lowe'a



Poprawka Gavina Lowe'a



Czy większy system też jest bezpieczny?

- Czy z faktu, że system z dwoma agentami i intruzem jest bezpieczny możemy coś wnioskować?
- Gavin Lowe proponuje dwa podejścia:
  - Dowodzić, że system niezależnie od rozmiaru jest bezpieczny, nie oglądając się na mały.
  - Dowieść, że jeśli duży były wadliwy to mały również.

Więcej uczciwych agentów

- Rozważmy atak na inicjatora protokołu *A.*

1.1	A -> I(B)	{Na, A}PK{B}
1.2	I(B) -> A	{Na, Nb, B}PK{A}
1.3	A -> I(B)	{Nb}PK{B}

- Intruz potrzebuje wysłać tylko jedną wiadomość (1.2), żeby to zrobić potrzebuje {Na, Nb, B}PK{A}.
- Z 1.1 nie może wywnioskować {Na, Nb, B}PK{A} ani Na.
- Potrzebny jest mu drugi przebieg (nazwijmy go 2), z którego mógłby się nauczyć potrzebnych wiadomości – stroną w nim musi być B, bo nikt inny nie rozszyfruje {Na, A}PK{B}.
- Intruz nie potrzebuje więcej informacji aby rozpocząć drugi przebieg, więc tylko te dwa są mu potrzebne.
- Jeśli potrafiłby się podszyć pod B w dużym systemie, mógłby to zrobić również w rozważanym przez nas.

Więcej uczciwych agentów

- Zajmijmy się teraz atakiem na odbiorcę *B.*

1.1	I(A) -> B	{Na, A}PK{B}
1.2	B -> A	{Na, Nb, B}PK{A}
1.3	I(A) -> B	{Nb}PK{B}

- Intruz potrzebuje wysłać wiadomości 1.1 i 1.2.
- Z 1.2 nie może poznać Nb, bo nie może rozszyfrować wiadomości.
- Musi powtórzyć zakodowaną wiadomość z 1.2 w innym biegu protokołu (nazwijmy go 2) jego inicjatorem musi być A (żeby mógł rozkodować 1.2).

2.1	A -> I(B)	{Na, A}PK{B}
2.2	I(B) -> A	{Na, Nb, B}PK{A}
2.3	A -> I(B)	{Nb}PK{B}

- {Na, A}PK{B} potrzebne w 2.1 zna z 1.2 (nie potrzebuje już żadnych więcej biegów) – jeżeli istnieje atak na duży system, jest on wykrywalny w małym.

Więcej intruzów

- Czy dwóch lub więcej intruzów współpracujących ze sobą może skuteczniej atakować nasz system?
- W wypadku naszego intruza jego swoboda jest na tyle duża, że może on symulować większą ilość napastników.
- Stąd wniosek: jeżeli istniałby atak w systemie z wieloma intruzami to istniałby też w systemie z jednym.

Modelowanie

- Strony protokołu
  - Stosunkowo nietrudne zadanie, ponieważ ich ruchy są dokładnie znane
  - Być może należy pomyśleć o dodaniu elementów umożliwiających lub ułatwiających weryfikację bezpieczeństwa

Modelowanie

- Intruz
  - Bardzo duży kłopot, bo musimy zamodelować najgorszego możliwego intruza (tzn. takiego, który będzie potrafił wykorzystać wszystkie możliwe formy ataku).
  - Taki napastnik potrafi podsłuchać i/lub przechwycić każdą wiadomość w systemie.
  - Następnym kłopotem jest reprezentacja wiedzy zdobywanej przez intruza, znowu musimy bardzo uważać, żeby czegoś nie przeoczyć.
  - Musimy pamiętać, że intruz się uczy i każdej chwili może wykorzystać całą zdobytą przez siebie wiedzę.

Modelowanie

- Bezpieczeństwo:
  - Znowu trudne zadanie, musimy wrazić w jakiś sposób, kiedy nasz system jest bezpieczny, a kiedy nie.
  - Przykład z naszego protokołu w LTL:
    - [] (([] !IniCommitAB) || (!IniCommitAB U ResRunningAB))
    - [] (([] !ResCommitAB) || (!ResCommitAB U IniRunningAB))

Weryfikacja

- Gavin Lowe znalazł błąd w protokole przy pomocy narzędzia FDR.
- Niestety to narzędzie stało się płatnie i nie mogę pokazać w jaki sposób Lowe zamodelował system i znalazł w nim błąd – dostępne jest wersja demo pozwalająca uruchamiać przykłady dostarczone z nią dostarczone, niestety nie ma wśród nich protokołu Needhama-Schroedera.
- Zaraz za to zobaczymy model w Promeli i weryfikację przy użyciu Spina.

Koniec

Dziękuję