VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

DATABÁZE SPECIFIKACÍ BEZPEČNOSTNÍCH PROTOKOLŮ

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE ZDENĚK PAPEŽ
AUTHOR

BRNO 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

DATABÁZE SPECIFIKACÍ BEZPEČNOSTNÍCH PROTOKOLŮ

SPECIFICATION DATABASE OF SECURITY PROTOCOLS

BAKALÁŘSKÁ PRÁCE

BACHELOR´S THESIS

AUTOR PRÁCE ZDENĚK PAPEŽ
AUTHOR

VEDOUCÍ PRÁCE ING. PAVEL OČENÁŠEK
SUPERVISOR

BRNO 2008

Zadání:

1. Seznamte se s problematikou bezpečnostních protokolů (BP), jejich vlastnostmi a využitím.
2. Zpracujte přehled kategorií jednotlivých BP. Uveďte seznam protokolů, které do jednotlivých kategorií patří.
3. Zpracujte dostupnou literaturu a u každého protokolu uveďte jeho specifikaci, vlastnosti a jeho detailní popis. Specifikaci protokolů proveďte pokud možno formálně a v univerzální formě tak, aby ji bylo možné využít externími aplikacemi.
4. Navrhněte systém pro uložení získaných informací ve formě databáze BP. Navrhněte webové prezentační rozhraní. Navrhněte dotazovací rozhraní systému tak, aby mohly ke specifikacím BP jednoduše přistupovat externí aplikace.
5. Implementujte funkce systému dle instrukcí vedoucího práce a naplňte databázi reálnými daty (specifikace BP).
6. Diskutujte získané znalosti a možnosti dalšího rozšíření databáze.

Licenční smlouva

Licenční smlouva je uložena v archívu Fakulty informačních technologií Vysokého učení technického v Brně.

Abstrakt

Tato bakalářská práce pojednává o bezpečnostních protokolech, které slouží jako prostředek pro vytvoření chráněné komunikace ve veřejné síti, Internetu. Bezpečnostních protokolů je celá řada. Proto jsem vybral jen pár nejznámějších protokolů a podrobně je analyzoval. Z dostupných zdrojů jsem shromáždil informace o protokolech, které vždy jednotně uvádím u každého z nich. Hlavním přínosem mé studie o bezpečnostních protokolech je prozkoumání množin, které se úzce vztahují k běhu bezpečnostního protokolu. Jedná se o množinu znalostí a množinu předpokladů jednotlivých subjektů, které se podílí na komunikaci pomocí daného protokolu. Tyto množiny pak uvádím v počátečním, cílovém a aktuálním stavu. Na závěr shrnuji všechny zpracované protokoly do přehledné tabulky, která je i s detaily všech protokolů převedena do počítačové databáze a veřejně přístupná na fakultních stránkách.

Klíčová slova

Bezpečnostní protokoly, specifikace, množina znalostí, množina předpokladů, útok, symetrické šifrování, asymetrické šifrování, důvěryhodný server, autentizace, distribuce klíče.

Abstract

This bachelor's thesis is about the security protocols which are necessary for safe communication in the Internet. There are various kinds of security protocols in use but this work covers only selection of most important of them. I have tried to gather all available information and work them out into uniform database which can be used for further processing. In my opinion, main aid of this work is precise description of knowledge base and prediction base of each protocol. Also all possible conditions and states are described. The end of part of this work sums up all concerned protocols into a user-friendly table which is used in web-side database accessible from the Internet.

Keywords

Security protocols, specification, knowledge base, prediction base, attack, symmetric encryption, asymmetric encryption, trusted server, authentication, key distribution.

Citace

Papež Zdeněk: Databáze specifikací bezpečnostních protokolů. Brno, 2008, bakalářská práce, FIT VUT v Brně.

Databáze specifikací bezpečnostních protokolů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Pavla Očenáška. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Zdeněk Papež
11. 5. 2008

Poděkování

Děkuji Ing. Pavlu Očenáškovi, vedoucímu bakalářské práce, který mi ochotně poskytl důležité informace a odborné rady při vytváření této práce.

© Zdeněk Papež, 2008.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah 1
1 Úvod 2
2 Základní pojmy 3
3 Zpracování protokolů 5
4 Rozdělení bezpečnostních protokolů 7
4.1 Protokoly s dvěma subjekty 7
4.1.1 Andrew Secure RPC 7
4.1.2 Diffie Helman 9
4.1.3 Schnorr's Protokol 11
4.2 Protokoly s více subjekty a se symetrickým šifrováním 13
4.2.1 Needham Schroeder Symmetric Key 13
4.2.2 Amended Needham Schroeder Symmetric Key 16
4.2.3 Denning-Sacco shared key 18
4.2.4 Lowe modified Denning-Sacco shared key 20
4.2.1 Kao Chow Authentication v.1 22
4.2.2 Kao Chow Authentication v.2 24
4.2.3 Wide Mouthed Frog 26
4.2.4 Yahalom 28
4.2.5 BAN simplified version of Yahalom 30
4.2.6 Lowe modified version of Yahalom 33
4.2.7 Otway Rees 35
4.2.8 KSL 37
4.2.9 Lowe modified KSL 41
4.2.10 Woo and Lam Pi 44
4.3 Protokoly s více subjekty a s asymetrickým šifrováním 46
4.3.2 Lowe's fixed version of Needham-Schroeder Public Key 49
4.3.3 TNM 51
5 Přehled bezpečnostních protokolů 54
6 Závěr 55
Literatura 56
Seznam obrázků 58
Seznam příloh 59
Příloha 1. – Manuál k databázi bezpečnostních protokolů 60

1 Úvod

Dnešní doba by se dala označit za dobu výpočetní techniky. Téměř všichni lidé využívají počítače – většina domácností vlastní počítač, většina firem je založena na informačních systémech. Používání Internetu se za posledních deset let stalo samozřejmostí. V minulosti nebylo třeba si lámat hlavu s bezpečností. Připojení k Internetu bylo výsadou armády, později hrstky nadšenců, kteří si vyměňovali touto cestou informace. Ovšem dnes Internet používáme k činnostem, které jsou ohledně bezpečnosti velice choulostivé. Přes Internet sdělujeme osobní údaje či spravujeme bankovní konta. Při všech těchto transakcích se vystavujeme nebezpečí zneužití naší osoby. Jelikož se v několika uplynulých letech rozmohla počítačová kriminalita, je bezpečnost počítačových sítí a počítačové komunikace velice aktuální otázkou.

Veškerá komunikace přes nechráněnou veřejnou síť prochází přes spoustu subjektů, a proto hrozí reálné nebezpečí útoku na naše údaje. Útočník může naše data zachytit, modifikovat nebo se nám může jevit jako věrohodná druhá strana, se kterou běžně komunikujeme. Jednou z možností, jak se tomuto nebezpečí vyhnout, je používání bezpečnostních protokolů. Na rozdíl od běžných protokolů zajišťují tyto protokoly bezpečnou komunikaci subjektů v nezabezpečené síti. Dva subjekty se pomocí bezpečnostního protokolu dohodnou na šifrování zpráv, které si budou vyměňovat, a nikdo neoprávněný nepřeruší jejich komunikaci. Bohužel to je jen teorie. Vždy se najde někdo, kdo daný bezpečnostní protokol obelstí a může ho zneužít. Proto se vymýšlí stále nové a nové bezpečnostní protokoly. Což vede k tomu, že nemůžeme označit žádnou komunikaci jako stoprocentně bezpečnou a chráněnou.

Ve své práci se zabývám jednotlivými bezpečnostními protokoly. Tyto protokoly zpracovávám jednotnou formou a vytvářím jedinečný český přehled bezpečnostních protokolů. Hlavní myšlenkou je vytvořit databázi těchto protokolů, která by byla přístupná na webu, a externí aplikace by byli schopné si stáhnout specifikaci jednotlivých protokolů pro provedení verifikace. Do této databáze se může pak dále přidávat další bezpečnostní protokoly a bude volně přístupná. V písemné části postupně v jednotlivých kapitolách vysvětluji základní pojmy, které souvisí s bezpečnostními protokoly, a způsob mého zpracování analýzy bezpečnostních protokolů. V další kapitole se nachází samotný základ mé práce neboli přehled zpracovaných bezpečnostních protokolů. Rozdělil jsem je podle počtu subjektů a způsobu šifrování zpráv. Hlavní důraz kladu na prozkoumání množin znalostí a předpokladů. Na závěr uvádím stručný seznam zpracovaných protokolů s důležitými znaky pro bezpečnostní protokoly.

2 Základní pojmy

K bezpečnostním protokolům, a počítačové komunikaci vůbec, patří několik pojmů, které v práci používám, a proto bych je v této kapitole rád uvedl a vysvětlil.

Bezpečnostní protokol

Bezpečnostní protokoly jsou, stejně jako všechny protokoly, soubor po sobě jdoucích dějů, které na sebe navazují a ovlivňují subjekty, které tímto protokolem komunikují. V těchto dějích si subjekty posílají zprávy, které jsou zašifrovány klíčem, který znají pouze zainteresované strany. Podle použitého klíče se bezpečnostní protokoly dělí na symetrické a asymetrické. Dále pak se mohou dělit podle funkce na autentizační protokoly a protokoly s distribucí klíče, ale většina bezpečnostních protokolů zaručuje obě funkce.

Autentizace

Autentizaci můžeme chápat jako několik po sobě jdoucích kroků, kterými si subjekty ověří identitu protějšího subjektu, se kterým komunikuje. Pokud bezpečnostní protokol obsahuje autentizační část, pak můžeme tento protokol označit za autentizační protokol. Samotná autentizace nejčastěji probíhá výměnou informací, které znají pouze zainteresované subjekty. Pokud je proces autentizace na dobré úrovni, pak je pro útočníka dost obtížné provést podvrh identity.

Distribuce klíče

Pro bezpečnou komunikaci v nechráněné síti je nezbytné, aby klienti komunikovali zašifrovanými zprávami. Proto je třeba zajistit zprostředkování čerstvých klíčů, které nikdo jiný nezná. Nejčastěji tuto funkci zajišťuje bezpečnostní protokol.V distribuci klíče může figurovat třetí strana, což bývá důvěryhodný server, kterému věří oba komunikující klienti. Třetí strana pak může fungovat jako KDC[1] a klíč vygenerovat nebo klíč vytvořený jedním klientem pouze předat dál, pak se jedna o KTC[2].

Nonce

Nonce je náhodně vygenerované číslo, či jiná informace, která se používá pro kontrolu identity klientů při výměně zpráv. Zároveň s časovými razítky zaručují čerstvost (*freshness*) zpráv a zejména pak distribuovaných hesel nebo lístků pro opakovanou autentizaci. Přestože je výraz nonce z angličtiny, používám toto převzaté slovo, jelikož český překlad „keksík“ považuji za nevhodný.

---

[1] KDC – Key Distribution Centre

[2] KDT – Key Transfer Centre

Symetrické šifrování

Symetrické šifrování je založeno na sdíleném tajemství, kdy oba subjekty, které šifrovaně komunikují, znají jeden symetrický klíč. Předpokládá se, že pro bezpečnost tento klíč znají pouze obě strany, popřípadě důvěryhodný server. Zprávu, kterou chce klient poslat, se zašifruje pomocí symetrického klíče a příjemce pak zprávu dešifruje stejným klíčem.

Asymetrické šifrování

Na rozdíl od symetrického šifrování, se pro asymetrického šifrování používá dvojice klíčů. Prvním z nich je veřejný klíč. Ten znají všechny subjekty, které chtějí komunikovat s vlastníkem tohoto klíče. Tento klíč je dostupný komukoliv a používá se k zašifrování zpráv. Subjekt, pro který je zašifrovaná zpráva určena, vlastní druhý klíč – soukromý klíč. Ten pak použije pro dešifrování zprávy mu určené.

Útok na zprávu

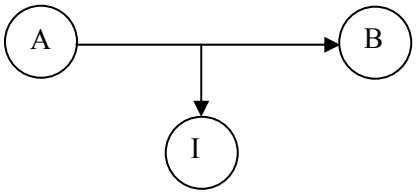
Většina bezpečnostních protokolů jsou nějakým způsobem napadnutelné. Útočníky můžeme rozdělit na pasivní, kteří pouze odposlouchávají komunikaci, a aktivní, kteří jsou schopni zasílat své zprávy. Podle toho můžeme mluvit o útoku přehráváním a útoku ze středu.

Útok přehráváním

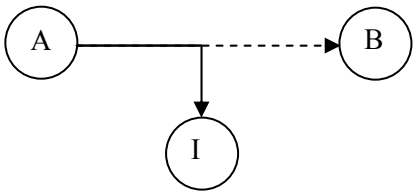
Při tomto útoku útočník odposlouchává komunikaci mezi subjekty. Tento záznam pak analyzuje a později může získané informace využít k podvržení identity. Proti tomuto útoku se do zpráv přidávají časová razítka a ty zaručí čerstvost zprávy, kterou subjekt přijímá.

Útok ze středu

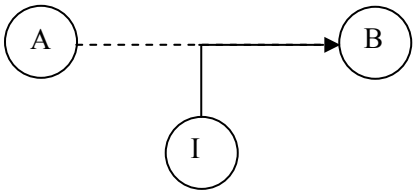
V tomto typu útoku se aktivní útočník vydává za oba zainteresované subjekty zároveň. Klient `A` si myslí, že komunikuje s klientem `B`, a naopak.



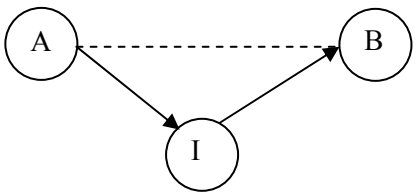
Obr. 3: Útok - Odposlouchávání



Obr. 1: Útok - Přerušení komunikace



Obr. 2: Útok - Podvržení identity



Obr. 4: Útok - Modifikace zprávy

3 Zpracování protokolů

V této kapitole budu popisovat způsob zpracování protokolů. Každý protokol, který jsem zpracoval, je popsán jednotným způsobem. Analýza obsahuje: vlastnosti, specifikaci protokolu, množinu počátečních znalostí subjektů, množinu počátečních předpokladů subjektů, množinu cílových znalostí, množinu aktuálních znalostí v průběhu protokolu, množinu aktuálních předpokladů v průběhu protokolu, nedostatky, autora a publikace.

Vlastnosti

Vlastnost protokolu obsahuje obecný princip funkce a komunikace protokolu, který je analyzován. Uvádím zajímavosti jednotlivých protokolů. Například jaké subjekty jsou použity nebo v jakém schématu jsou zapojeny. Pokud se jedná o upravenou verzi již staršího protokolu, tak zmiňuji provedené změny. Pro názornost uvádím obrázek, na kterém lze vidět v jakém pořadí a jak probíhá komunikace daného bezpečnostního protokolu.

Specifikace

Specifikace protokolu je soubor pravidel v určitém pořadí, kterými si subjekty komunikace vyměňují informace. Tato pravidla jsou zapsána pomocí jednotné syntaxe. Také definují jednotlivé subjekty protokolu a jiné informace, jako například časová razítka, nonce a klíče, která jsou použita při komunikaci. Zápis syntaxe je následující:

`A, B, S`	:	Subjekty komunikace
`M, N`	:	Nonce
`T, L`	:	Časové razítko
`Kab`	:	Symetrický klíč mezi subjekty `A` a `B`
`PKs`	:	Veřejný klíč serveru `S`
`SKs`	:	Soukromý klíč serveru `S`
`Succ(), dec()`	:	Funkce
`K', N'`	:	Nově zavedené klíče či nonce
`{…}K`	:	Šifrovaná zpráva symetrickým klíčem

Zápis pravidla:

```
Číslo pravidla. Odesilatel -> příjemce : obsah zprávy
```

Publikace

Publikace, ve které byl protokol prezentován a definován pro odborný svět nebo ve které je uveden úspěšný útok.

Množina počátečních znalostí subjektů

Každý subjekt, který komunikuje daným protokolem, zná před navázáním komunikace jisté znalosti, které můžu označit za množinu počátečních znalostí. Například sdílený symetrický klíč pro dva subjekty.

Množina počátečních předpokladů subjektů

Pro správnou komunikaci dvou subjektů pomocí daného protokolu je nezbytné, aby každý subjekt komunikace znal množinu znalostí ostatních subjektů. Tudíž pokud například server posílá zašifrovanou zprávu klientovi, předpokládá, že klient má příslušný klíč pro dešifrování posílané zprávy.

Množina cílových znalostí

Všechny protokoly mají určitý cíl. Tento cíl je vymezen množinou cílových znalostí jednotlivých subjektů. U bezpečnostních protokolů je většinou kýženým výsledkem znalost nového relačního klíče nebo potvrzení identity subjektů. Speciálním případem může být znalost tzv. lístku, pro autentizaci klienta.

Množina aktuálních znalostí v průběhu protokolu

Po každém provedení jednoho pravidla protokolu se změní množina znalostí zainteresovaných subjektů. Po provedení posledního kroku komunikace by měla tato množina obsahovat podmnožinu cílů daného protokolu.

Množina aktuálních předpokladů v průběhu protokolu

Stejně jako množina znalostí se množina předpokladů mění při provádění jednotlivých pravidel. Sledování těchto dvou množin vede k podrobné analýze protokolů. Porovnání cílových znalostí a předpokladů se skutečností také vede k formální verifikaci, což je oblast, která úzce navazuje na toto téma.

Nedostatky

V této podkapitole uvádím nedostatky bezpečnostních protokolů. Jedná se o útoky, které byly někým provedeny a uvedeny veřejnosti. Tyto útoky se snažím vysvětlit a názorně uvést pomocí specifikace, kde se subjekt útočník označuje písmenem `I`. Při podvržení identity pak `Ia` označuje útočníka, který se vydává za klienta `A`.

Autor

Autor nebo skupina autorů, kteří vymysleli uvedený bezpečnostní protokol a publikovali ho v uvedeném roce.

4 Rozdělení bezpečnostních protokolů

Bezpečnostních protokolů je celá řada. Pro vetší přehlednost jsem protokoly rozdělil do skupin podle počtu subjektů, které v protokolu figurují, a způsobu šifrování komunikace mezi nimi.Tímto rozdělením vznikly tři skupiny: protokoly s dvěma subjekty, protokoly se třemi subjekty, které jsem dále rozdělil na protokoly využívající symetrické a asymetrické šifrování zpráv.

4.1 Protokoly s dvěma subjekty

Protokoly, které probíhají jen mezi dvěma subjekty, neboli nevyužívají třetí důvěryhodnou stranu. Tyto protokoly jsou většinou jednoduché a probíhají v několika krocích vzájemné komunikace. Nejsou však až tak rozšířené. Jedná se o protokoly, které zajišťují pouze autentizaci nebo jen distribuci klíčů. Jako zástupce této rodiny uvádím Andrew Secure RPC, Diffie helman a Shnorr's Protokol.

4.1.1 Andrew Secure RPC

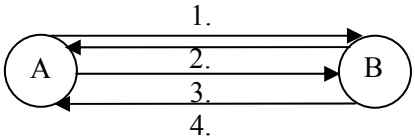
4.1.1.1 Vlastnosti

Protokol Andrew Secure RPC [1] patří k protokolům, které k vzájemné autentizace dvou klientů nepotřebují třetí stranu. Tento protokol předpokládá, že obě strany znají společný klíč. Tímto klíčem šifrují zprávy, které si vzájemně posílají, a při každé nově navázané komunikaci vzniká nový symetrický klíč, kterým budou komunikovat v budoucnu. Nonces `Na` a `Nb` a jednosměrný algoritmus, který tyto nonces nezvratně změní, slouží k autentizaci klientů. Novou nonce `N'b` si subjekt `A` uchová pro nové spojení, aby mohl ověřit identitu subjektu `B`.

4.1.1.2 Specifikace

```
A, B :          Subjekty
Na, Nb, N'b :   Nonces
succ() :        nonce  -> nonce
Kab, K'ab :     Klíče

1.  A->B :  A, {Na}Kab
2.  B->A :  {succ(Na), Nb}Kab
3.  A->B :  {succ(Na)}Kab
4.  B->A :  {K'ab, N'b}Kab
```



Obr. 5: Schéma komunikace - Andrew Secure RPC

A – klient
B – klient (klient A chce komunikovat s klientem B)
Kab – klíč pro komunikaci mezi klientem A a B
K'ab – nový klíč pro komunikaci mezi klientem A a B
succ() – změna nonce podle daného algoritmu

4.1.1.3 Množina počátečních znalostí subjektů

A	A, B, Kab, Nb
B	A, B, Kab, Nb

4.1.1.4 Množina počátečních předpokladů subjektů

A	→B: A, Kab, Nb
B	→A: B, Kab, Nb

4.1.1.5 Množina cílových znalostí subjektů

A	K'ab, N'b
B	K'ab, N'b

4.1.1.6 Množina aktuálních znalostí v průběhu protokolu

1	A	A, B, Kab, **Na**, Nb
	B	A, B, Kab, **Na**, Nb
2	A	A, B, Kab, Na, Nb, **succ(Na)**
	B	A, B, Kab, Na, Nb, **succ(Na)**
3	A	A, B, Kab, Na, Nb, succ(Na), **succ(Na)**
	B	A, B, Kab, Na, Nb, succ(Na), **succ(Na)**
4	A	A, B, Kab, **K'ab**, Na, Nb, succ(Na), succ(Na), **N'b**
	B	A, B, Kab, **K'ab**, Na, Nb, succ(Na), succ(Na), **N'b**

4.1.1.7 Množina aktuálních předpokladů v průběhu protokolu

1	A	→B: A, Kab, **Na**, Nb
	B	→A: B, Kab, **Na**, Nb
2	A	→B: A, Kab, Na, Nb, **succ(Na)**
	B	→A: B, Kab, Na, Nb, **succ(Na)**
3	A	→B: A, Kab, Na, Nb, succ(Na), **succ(Na)**
	B	→A: B, Kab, Na, Nb, succ(Na), **succ(Na)**
4	A	→B: A, Kab, **K'ab**, Na, Nb, succ(Na), succ(Na), **N'b**
	B	→A: B, Kab, **K'ab**, Na, Nb, succ(Na), succ(Na), **N'b**

4.1.1.8 Nedostatky

Slabým místem protokolu Andrew Secure RPC je konečný bod komunikace kde subjekt B zasílá nový symetrický klíč subjektu A. Ten nemůže tušit, zda je nový klíč opravdu „čerstvý“. Proto může útočník odchytit zprávu s novým klíčem a použít ho v době, kdy je již neplatný, což ale subjekt A nepozná [2].

4.1.1.9 Autor

M. Satyanarayanan 1987

4.1.1.10 Publikace

Reference na protokol: [1].

Útoky na tento protokol: [2].

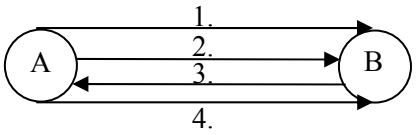
4.1.2 Diffie Helman

4.1.2.1 Vlastnosti

Diffie Helman [3] je bezpečnostní protokol, který zaručuje výměnu nového symetrického klíče (číslo vygenerované funkcí `one()`) mezi dvěma klienty. Pro šifrovanou komunikaci využívá funkci `kap`, která je vysvětlena níže. Tento algoritmus musí znát obě strany, jelikož právě tato funkce hraje v Diffie Helman protokolu roli klíče.

4.1.2.2 Specifikace

```
A, B:            Subjekty
P, G, Xa, Xb :   Čísla
one() :          ->číslo
Kap :            Číslo, číslo, číslo -> číslo
```



Obr. 6: Schéma komunikace - Diffie helman

```
1.  A->B :  P, G
2.  A->B :  kap(P, G, Xa)
3.  B->A :  kap(P, G, Xb)
4.  A->B :  {one()}kap(P, kap(P, G, Xb), Xa)
```

`A` – klient
`B` – klient (klient `A` chce komunikovat s klientem `B`)
`P` – primární číslo
`G` – číslo, platí `G < P`
`one` – funkce, která vygeneruje číslo
`kap` – funkce, pro kterou platí:

```
kap(P, kap(P, G, Y), X) = kap(P, kap(P, G, X), Y)
kap(P, X, Y) = exp(X, Y) mod P.
```

4.1.2.3 Množina počátečních znalostí subjektů

A	`A, B`
B	`A, B`

4.1.2.4 Množina počátečních předpokladů subjektů

A	`→B: A, B`
B	`→A: A, B`

4.1.2.5 Množina cílových znalostí subjektů

A	one()
B	one()

4.1.2.6 Množina aktuálních znalostí v průběhu protokolu

1	A	A, B, **P, G**
	B	B, S, **P, G**
2	A	A, B, P, G, **Xa, kap(P, G, Xa)**
	B	B, S, P, G, **kap(P, G, Xa)**
3	A	A, B, P, G, Xa, kap(P, G, Xa), **kap(P, G, Xb)**
	B	B, S, P, G, **Xb**, kap(P, G, Xa), **kap(P, G, Xb)**
4	A	A, B, P, G, Xa, kap(P, G, Xa), kap(P, G, Xb), **one()**
	B	B, S, P, G, Xb, kap(P, G, Xa), kap(P, G, Xb), **one()**

4.1.2.7 Množina aktuálních předpokladů v průběhu protokolu

1	A	→B: A, B, **P, G**
	B	→A: A, B, **P, G**
2	A	→B: A, B, P, G, **kap(P, G, Xa)**
	B	→A: A, B, P, G, **kap(P, G, Xa)**
3	A	→B: A, B, P, G, kap(P, G, Xa), **kap(P, G, Xb)**
	B	→A: A, B, P, G, kap(P, G, Xa), **kap(P, G, Xb)**
4	A	→B: A, B, P, G, kap(P, G, Xa), kap(P, G, Xb), **one()**
	B	→A: A, B, P, G, kap(P, G, Xa), kap(P, G, Xb), **one()**

4.1.2.8 Nedostatky

Protokol Diffie helman je napadnutelný podvržením identity, jelikož ani jedna strana si nemůže ověřit identitu té druhé, jelikož nedochází k žádné autentizaci. Útočník tak může nahradit v komunikaci subjekt A (jak je uvedeno níže) i B [4].

```
i.1   Ia → B :        P, G
i.2   Ia → B :        kap(P, G, Xi)
i.3   B  → Ia :       kap(P, G, Xb)
i.4   Ia → B :        {one()}kap(P, kap(P, G, Xb), Xi)
```

4.1.2.9 Autor

W. Diffie a M. Helman 1978

4.1.2.10 Publikace

Reference na protokol: [3].

Útoky na tento protokol: [4].

4.1.3 Schnorr's Protokol

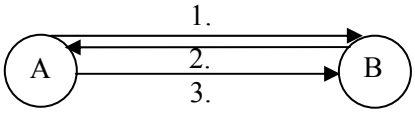
4.1.3.1 Vlastnosti

Schnorr's Protokol [5] je mechanizmus, který zaručuje autentizaci dvou subjektů, aniž by subjekty sdílely společné tajemství. Pro asymetrické šifrování slouží soukromý a veřejný klíč klienta `A`. Po výměně poslední zprávy se klient `B` ujistí o identitě klienta `A` tím, že zkontroluje, zda platí `exp(g,r) = a × exp(Pa,Nb)`.

4.1.3.2 Specifikace

```
A, B :                  Subjekty
Na, Nb :                Nonces
SKa :                   Soukromý klíč
PKa = exp(g, SKa) :     Veřejný klíč

1.  A->B :  a
2.  B->A :  Nb
3.  A->B :  r
```



Obr. 7: Schéma komunikace - Schorr's protokol

`A` - klient
`B` – klient (klient `A` chce komunikovat s klientem `B`)
`a` – pro `a` platí: a = exp(g, Na)
`r` – pro `r` platí: r = Na + Nb × SKa
+ – sčítání
× – násobení
`exp` – funkce exponent
Pro výraz a × exp(PKa,Nb) platí:

$$\begin{aligned} a \times \exp(PKa,Nb) &= \exp(g,Na) \times \exp(\exp(g,SKa),Nb) \\ &= \exp(g,Na) \times \exp(g,SKa \times Nb) \\ &= \exp(g,Na + SKa \times Nb) \\ &= \exp(g, r) \end{aligned}$$

4.1.3.3 Množina počátečních znalostí subjektů

A	`A, B, SKa, g`
B	`A, B, PKa, g`

4.1.3.4 Množina počátečních předpokladů subjektů

A	`→B: A, B, PKa, g`
B	`→A: A, B, SKa, g`

4.1.3.5 Množina cílových znalostí subjektů

Subjekt B zkontroluje, zda pro příchozí výraz platí : exp(g, r) = a × exp(PKa,Nb).

4.1.3.6 Množina aktuálních znalostí v průběhu protokolu

1	A	A, B, SKa, Na, **a**, g
	B	A, B, PKa, **a**, g
2	A	A, B, SKa, Na, **Nb**, a, g
	B	A, B, PKa, **Nb**, a, g
3	A	A, B, SKa, Na, Nb, a, **r**, g
	B	A, B, PKa, Nb, a, **r**, g

4.1.3.7 Množina aktuálních předpokladů v průběhu protokolu

1	A	→B: A, B, PKa, **a**, g
	B	→A: A, B, SKa, **a**, g
2	A	→B: A, B, PKa, **Nb**, a, g
	B	→A: A, B, SKa, **Nb**, a, g
3	A	→B: A, B, PKa, Nb, a, **r**, g
	B	→A: A, B, SKa, Nb, a, **r**, g

4.1.3.8 Autor

C. P. Schnorr 1991

4.1.3.9 Publikace

Protokol je popsán v publikaci: [5].

4.2 Protokoly s více subjekty a se symetrickým šifrováním

Tyto protokoly využívají k předání relačních klíčů třetí subjekt, neboli třetí stranu, důvěryhodný server. Jedná se převážně o protokoly, které zaručí výměnu symetrického klíče a následně pak provedení autentizace. K nejznámějším protokolům, jež jsem zařadil do této kategorie, patří: Needham Schroeder, Denning Sacco, Yahalom, Otway Rees, KSL a další.

4.2.1 Needham Schroeder Symmetric Key

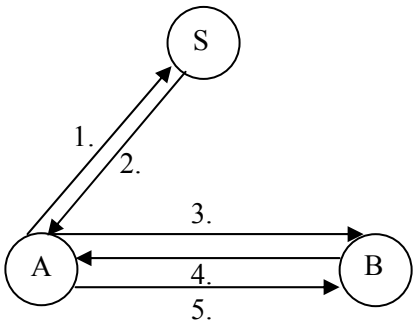
4.2.1.1 Vlastnosti

Needham Shroeder [6] patří mezi jednoduché protokoly, proto je často používán jako příklad verifikačních metod. Protokol Needham Shroeder existuje v asymetrické i symetrické podobě, kterou zde uvádím. Pomocí tohoto protokolu chce subjekt `A` navázat šifrovanou komunikaci se subjektem `B`. Pro navázání komunikace se využije třetí strany, která sdělí subjektu `A` vygenerovaný symetrický klíč pro šifrování zpráv mezi subjekty `A` a `B`. Body čtyři a pět v komunikaci mezi subjekty slouží pro autentizaci.

4.2.1.2 Specifikace

```
A, B, S:        Subjekty
Na, Nb:         Nonce
Kas, Kbs, Kab:  Klíče
Dec:            Nonce -> Nonce

1. A->S: A, B, Na
2. S->A: {Na, B, Kab, {Kab, A}Kbs}Kas
3. A->B: {Kab,A}Kbs
4. B->A: {Nb}Kab
5. A->B: {dec(Nb)}Kab
```



Obr. 8: Schéma komunikace - Needham Schroeder Symetric Key

`A` – klient
`B` – klient (klient A chce komunikovat s klientem B)
`S` – server (server umožní komunikaci. Předá klientovi a klíč pro komunikaci)
`Kas` – klíč pro komunikaci mezi klientem A a serverem
`Kbs` – klíč pro komunikaci mezi klientem B a serverem
`Kab` – klíč pro komunikaci mezi klientem A a klientem B
`Dec` – dekrementace nonce

4.2.1.3 Množina počátečních znalostí subjektů

A	Kas, A, S, B
B	Kbs, B, S
S	Kas, Kbs, S, A, B,

4.2.1.4 Množina počátečních předpokladů subjektů

A	→B: Kbs, S →S: Kas, A, B
B	→S: Kbs, B
S	→A: Kas, S, B →B: Kbs, S

4.2.1.5 Množina cílových znalostí

A	Kab
B	Kab, dec(Nb)

4.2.1.6 Množina aktuálních znalostí v průběhu protokolu

1	A	Kas, B, S, **Na**
	B	Kbs, S
	S	Kas, Kbs, A, B, **Na**
2	A	Kas, **Kab**, B, S, Na, **{Kab, A}Kbs**
	B	Kbs, S
	S	Kas, Kbs, **Kab**, A, B, Na
3	A	Kas, Kab, B, S, Na, {Kab, A}Kbs
	B	Kbs, **Kab**, S, **A**
	S	Kas, Kbs, Kab, A, B, Na
4	A	Kas, Kab, B, S, Na, **Nb**, {Kab, A}Kbs
	B	Kbs, Kab, S, A, **Nb**
	S	Kas, Kbs, Kab, A, B, Na
5	A	Kas, Kab, B, S, Na, Nb, {Kab, A}Kbs, **dec(Nb)**
	B	Kbs, Kab, S, A, Nb, **dec(Nb)**
	S	Kas, Kbs, Kab, A, B, Na

4.2.1.7 Množina aktuálních předpokladů v průběhu protokolu

1	A	→B: Kbs, S, B →S: Kas, A, B, S, **Na**
	B	→S: Kbs, B, S
	S	→A: Kas, A, B, S, **Na** →B: Kbs, B, S
2	A	→B: Kbs, S, B →S: Kas, **Kab**, A, B, S, Na, **{Kab, A}Kbs**
	B	→S: Kbs, B, S
	S	→A: Kas, **Kab**, A, B, S, Na, **{Kab, A}Kbs** →B: Kbs, B, S

3	A	→B: Kbs, **A**, S, B, **{Kab, A}Kbs** →S: Kas, Kab, A, B, S, Na, {Kab, A}Kbs
	B	→A: **A, B, {Kab, A}Kbs** →S: Kbs, B, S
	S	→A: Kas, Kab, A, B, S, Na, {Kab, A}Kbs →B: Kbs, B, S
4	A	→B: Kbs, A, S, B, **Nb**, {Kab, A}Kbs →S: Kas, Kab, A, B, S, Na, {Kab, A}Kbs
	B	→A: A, B, **Nb**, {Kab, A}Kbs →S: Kbs, B, S
	S	→A: Kas, Kab, A, B, S, Na, {Kab, A}Kbs →B: Kbs, B, S
5	A	→B: Kbs, A, S, B, Nb, {Kab, A}Kbs, **dec(Nb)** →S: Kas, Kab, A, B, S, Na, {Kab, A}Kbs
	B	→A: A, B, Nb, {Kab, A}Kbs, **dec(Nb)** →S: Kbs, B, S
	S	→A: Kas, Kab, A, B, S, Na, {Kab, A}Kbs →B: Kbs, B, S

4.2.1.8 Nedostatky

Protokol Needham Schroeder je náchylný na útoky typu přerušení komunikace, odposlechu (sekce `i`) a dále podvržení identity (sekce `ii`) [7]. Útočník `I` se vydává za subjekt `B` a tím odhalí klíč `Kab` a může komunikovat se subjektem `B`, který podvod neodhalí.

```
 i.1  A  -> S  :     A, B, Na
 i.2  S  -> A  :     {Na, B, Kab, {Kab, A}Kbs }Kas
 i.3  A  -> Ib :     {Kab,A}Kbs
ii.1  Ia -> B  :     {Kab,A}Kbs
ii.2  B  -> Ia :     {Nb}Kab
ii.3  Ia -> B  :     {dec(Nb)}Kab
```

4.2.1.9 Autor

Roger Needham a Michael Schroeder 1978

4.2.1.10 Publikace

Protokol byl uveden: [6].

Útoky na protokol Needham Schroeder Symmetric Key jsou uvedeny: [7].

4.2.2 Amended Needham Schroeder Symmetric Key

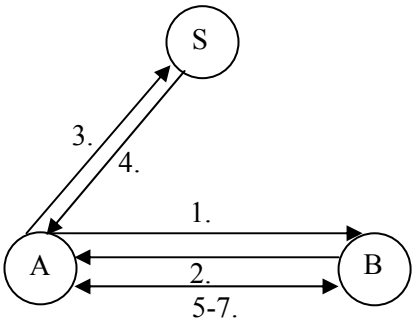
4.2.2.1 Vlastnosti

V roce 1987 autoři protokolu Needham Schroeder Symmetric Key tento protokol upravili a opravili a vznikl protokol Amended Needham Schroeder Symmetric Key [8]. Tento protokol opět zaručuje distribuci symetrického klíče oběma klientům přes důvěryhodný server a obousměrnou autentizaci. Samotná změna tkví v přidání nonce `Nb` do zprávy, která obsahuje nový symetrický klíč relace. To zabrání předchozím útokům typu přerušení komunikace a podvržení komunikace, na které byla náchylná předešlá verze tohoto protokolu.

4.2.2.2 Specifikace

```
A, B, S:        Subjekty
Na, Nb:         Nonce
Kas, Kbs, Kab:  Klíče
dec:            Nonce -> Nonce

1.  A->B:  A
2.  B->A:  {A, Nb}Kbs
3.  A->S:  A, B, Na, {A, Nb}Kbs
4.  S->A:  {Na, B, Kab, {Kab, Nb, A}Kbs}Kas
5.  A->B:  {Kab, Nb, A}Kbs
6.  B->A:  {Nb}Kab
7.  A->B:  {dec(Nb)}Kab
```



Obr. 9: Schéma komunikace - Amended Needham Schroeder Symetric Key

`A` – klient
`B` – klient (klient A chce komunikovat s klientem B)
`S` - server (server umožní komunikaci. Předá klientovi a klíč pro komunikaci)
`Kas` – klíč pro komunikaci mezi klientem A a serverem
`Kbs` – klíč pro komunikaci mezi klientem B a serverem
`Kab` – klíč pro komunikaci mezi klientem A a klientem B
`dec` – dekrementace nonce

4.2.2.3 Množina počátečních znalostí subjektů

A	A, B, S, Kas
B	B, S, Kbs
S	A, B, S, Kas, Kbs

4.2.2.4 Množina počátečních předpokladů subjektů

A	→B: B, S, Kbs →S: A, B, S, Kas
B	→S: B, S, Kbs
S	→A: A, B, S, Kas →B: B, S, Kbs

4.2.2.5 Množina cílových znalostí

A	Kab
B	Kab, dec(Nb)

4.2.2.6 Množina aktuálních znalostí v průběhu protokolu

1	A	A, B, S, Kas
	B	**A**, B, S, Kbs
	S	A, B, S, Kas, Kbs
2	A	A, B, S, Kas, **{A, Nb}Kbs**
	B	A, B, S, Kbs, **Nb**
	S	A, B, S, Kas, Kbs
3	A	A, B, S, Kas, Na, {A, Nb}Kbs
	B	A, B, S, Kbs, Nb
	S	A, B, S, Kas, Kbs, **Na**, **Nb**
4	A	A, B, S, **Kab**, Kas, Na, {A, Nb}Kbs, **{Kab, Nb, A}Kbs**
	B	A, B, S, Kbs, Nb
	S	A, B, S, **Kab**, Kas, Kbs, Na, Nb
5	A	A, B, S, Kab, Kas, Na, {A, Nb}Kbs, {Kab, Nb, A}Kbs
	B	A, B, S, **Kab**, Kbs, Nb
	S	A, B, S, Kab, Kas, Kbs, Na, Nb
6	A	A, B, S, Kab, Kas, Na, **Nb**, {A, Nb}Kbs, {Kab, Nb, A}Kbs
	B	A, B, S, Kab, Kbs, Nb
	S	A, B, S, Kab, Kas, Kbs, Na, Nb
7	A	A, B, S, Kab, Kas, Na, Nb, {A, Nb}Kbs, {Kab, Nb, A}Kbs, **dec(Nb)**
	B	A, B, S, Kab, Kbs, Nb, **dec(Nb)**
	S	A, B, S, Kab, Kas, Kbs, Na, Nb

4.2.2.7 Množina aktuálních předpokladů v průběhu protokolu

1	A	→B: **A**, B, S, Kbs →S: A, B, S, Kas
	B	→A: **A**, **B** →S: B, S, Kbs
	S	→A: A, B, S, Kas →B: B, S, Kbs
2	A	→B: A, B, S, Kbs, **{A, Nb}Kbs** →S: A, B, S, Kas
	B	→A: A, B, **{A, Nb}Kbs** →S: B, S, Kbs
	S	→A: A, B, S, Kas →B: B, S, Kbs
3	A	→B: A, B, S, Kbs, {A, Nb}Kbs →S: A, B, S, Kas, **Na**, **{A, Nb}Kbs**
	B	→A: A, B, {A, Nb}Kbs →S: B, S, Kbs
	S	→A: A, B, S, Kas, **Na**, **{A, Nb}Kbs** →B: B, S, Kbs
4	A	→B: A, B, S, Kbs, {A, Nb}Kbs →S: A, B, S, **Kab**, Kas, Na, {A, Nb}Kbs, **{Kab, Nb, A}Kbs**
	B	→A: A, B, {A, Nb}Kbs →S: B, S, Kbs
	S	→A: A, B, S, Na, **Kab**, Kas, {A, Nb}Kbs, **{Kab, Nb, A}Kbs** →B: B, S, Kbs

5	A	→B: A, B, S, Kbs, {A, Nb}Kbs, **{Kab, Nb, A}Kbs** →S: A, B, S, Kab, Kas, Na, {A, Nb}Kbs, {Kab, Nb, A}Kbs
	B	→A: A, B, Kab, {A, Nb}Kbs, **{Kab, Nb, A}Kbs** →S: B, S, Kbs
	S	→A: A, B, S, Na, Kab, Kas, {A, Nb}Kbs, {Kab, Nb, A}Kbs →B: B, S, Kbs
6	A	→B: A, B, S, **{Nb}Kab**, Kab, Kbs, {A, Nb}Kbs, {Kab, Nb, A}Kbs →S: A, B, S, Kab, Kas, Na, {A, Nb}Kbs, {Kab, Nb, A}Kbs
	B	→A: A, B, Kab, **{Nb}Kab**, {A, Nb}Kbs, {Kab, Nb, A}Kbs →S: B, S, Kbs
	S	→A: A, B, S, Na, Kab, Kas, {A, Nb}Kbs, {Kab, Nb, A}Kbs →B: B, S, Kbs
7	A	→B: A, B, S, Nb, Kab, Kbs, {A, Nb}Kbs, {Kab, Nb, A}Kbs,**{dec(Nb)}Kab** →S: A, B, S, Kab, Kas, Na, {A, Nb}Kbs, {Kab, Nb, A}Kbs
	B	→A: A, B, Kab, Nb, {A, Nb}Kbs, {Kab, Nb, A}Kbs, **{dec(Nb)}Kab** →S: B, S, Kbs
	S	→A: A, B, S, Na, Kab, Kas, {A, Nb}Kbs, {Kab, Nb, A}Kbs →B: B, S, Kbs

4.2.2.8 Autor

Roger Needham a Michael Schroeder 1987

4.2.2.9 Publikace

Protokol byl uveden v publikaci: [8].

4.2.3 Denning-Sacco shared key

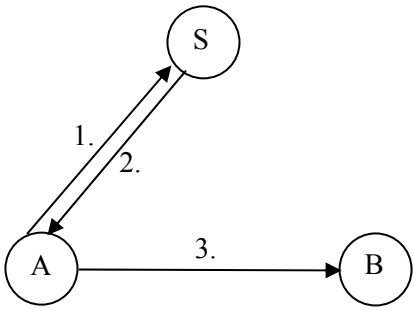
4.2.3.1 Vlastnosti

Denning-Sacco shared key [7] je protokol, který vychází z protokolu Needham Schroeder Symmetric Key. Pro distribuci symetrického klíče se také používá třetí strana. Změna nastává v nahrazení nonce časovým razítkem, které zaručuje aktuálnost distribuovaného klíče.

4.2.3.2 Specifikace

```
A, B, S:          Subjekty
T :               Časové razítko
Kas, Kbs, Kab :   Klíče

1.  A->S :  A, B
2.  S->A :  {B, Kab, T, {Kab, A, T}Kbs}Kas
3.  A->B :  {Kab,A, T}Kbs
```



Obr. 10: Schéma komunikace - Denning-Sacco shared key

A – klient
B – klient (klient A chce komunikovat s klientem B)
S – server, distribuce klíčů
Kab – klíč pro komunikaci mezi klientem A a B
Kas – klíč pro komunikaci mezi klientem A a serverem S
Kab – klíč pro komunikaci mezi klientem B a serverem S

4.2.3.3 Množina počátečních znalostí subjektů

A	A, B, S, Kas
B	B, S, Kbs
S	A, B, S, Kas, Kbs

4.2.3.4 Množina počátečních předpokladů subjektů

A	→B: B, S →S: A, B, S, Kas
B	→S: B, S, Kbs
S	→A: A, S, Kas →B: B, S, Kbs

4.2.3.5 Množina cílových znalostí subjektů

A	Kab
B	Kab

4.2.3.6 Množina aktuálních znalostí v průběhu protokolu

1	A	A, B, S, Kas
	B	B, S, Kbs
	S	A, B, S, Kas, Kbs
2	A	A, B, S, **Kab**, Kas, **T**, **{Kab, A, T}Kbs**
	B	B, S, Kbs
	S	A, B, S, **Kab**, Kas, Kbs, **T**
3	A	A, B, S, Kab, Kas, T, {Kab, A, T}Kbs
	B	**A**, B, S, **Kab**, Kbs, **T**
	S	A, B, S, Kab, Kas, Kbs, T

4.2.3.7 Množina aktuálních předpokladů v průběhu protokolu

1	A	→B: B, S →S: A, B, S, Kas
	B	→S: B, S, Kbs
	S	→A: A, B, S, Kas →B: B, S, Kbs
2	A	→B: B, S →S: A, B, S, **Kab**, Kas, **T**, **{Kab, A, T}Kbs**
	B	→S: B, S, Kbs
	S	→A: A, B, S, **Kab**, Kas, **T**, **{Kab, A, T}Kbs** →B: B, S, Kbs
3	A	→B: A, B, S, **{Kab, A, T}Kbs** →S: A, B, S, Kab, Kas, T, {Kab, A, T}Kbs
	B	→A: **A**, **B**, **{Kab, A, T}Kbs** →S: B, S, Kbs
	S	→A: A, B, S, Kab, Kas, T, {Kab, A, T}Kbs →B: B, S, Kbs

4.2.3.8 Nedostatky

Tento protokol je napadnutelný podvržením identity, kdy se útočník vydává za subjekt `A` v třetím bodu komunikace při sdělování vzájemného klíče [9]. Subjekt `B` se domnívá, že došlo ke změně symetrického klíče.

```
i.3   Ia -> B :        {Kab,A, T}Kbs
```

4.2.3.9 Autor

Dorothy E. Denning a Giovanni Maria Sacco 1981

4.2.3.10 Publikace

Reference na protokol: [7].

Útoky na tento protokol: [9].

4.2.4 Lowe modified Denning-Sacco shared key

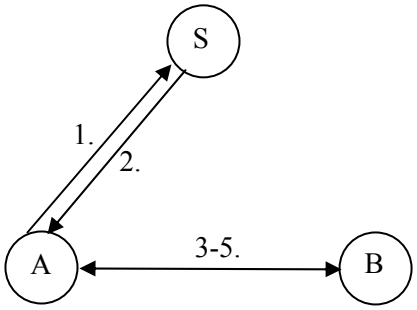
4.2.4.1 Vlastnosti

Gavin Lowe v roce 1997 upravil protokol Denning-Sacco shared key a tím vznikl protokol Lowe modified Denning-Sacco shared key [9]. K původním třem krokům přidal další dva, které zajišťují zpětné ověření identity takzvaným potřesením ruky (*handshake*), kdy si subjekty vymění nonce. Tím zabrání útokům, které byly účinné proti původní verzi protokolu.

4.2.4.2 Specifikace

```
A, B, S:          Subjekty
T :               Časové razítko
Kas, Kbs, Kab :   Klíče
Nb :              Nonce
dec :             nonce -> nonce

1.   A -> S :   A, B
2.   S -> A :   {B, Kab, T, {Kab, A, T}Kbs}Kas
3.   A -> B :   {Kab,A, T}Kbs
4.   B -> A :   {Nb}Kab
5.   A -> B :   {dec(Nb)}Kab
```



Obr. 11: Schéma komunikace - Lowe modified Denning-Sacco shared key

`A` – klient
`B` – klient (klient `A` chce komunikovat s klientem `B`)
`S` – server, distribuce klíčů
`Kab` – klíč pro komunikaci mezi klientem `A` a `B`
`Kas` – klíč pro komunikaci mezi klientem `A` a serverem `S`
`Kab` – klíč pro komunikaci mezi klientem `B` a serverem `S`
`dec` – dekrementace nonce

4.2.4.3 Množina počátečních znalostí subjektů

A	A, B, S, Kas
B	B, S, Kbs
S	A, B, S, Kas, Kbs

4.2.4.4 Množina počátečních předpokladů subjektů

A	→B: B, S →S: A, B, S, Kas
B	→S: B, S, Kbs
S	→A: A, S, Kas →B: B, S, Kbs

4.2.4.5 Množina cílových znalostí subjektů

A	Kab
B	Kab, dec(Nb)

4.2.4.6 Množina aktuálních znalostí v průběhu protokolu

1	A	A, B, S, Kas
	B	B, S, Kbs
	S	A, B, S, Kas, Kbs
2	A	A, B, S, **Kab**, Kas, **T**, **{Kab, A, T}Kbs**
	B	B, S, Kbs
	S	A, B, S, **Kab**, Kas, Kbs, **T**
3	A	A, B, S, Kab, Kas, T, {Kab, A, T}Kbs
	B	A, B, S, **Kab**, Kbs, **T**
	S	A, B, S, Kab, Kas, Kbs, T
4	A	A, B, S, Kab, Kas, T, **Nb**, {Kab, A, T}Kbs
	B	A, B, S, Kab, Kbs, T, **Nb**
	S	A, B, S, Kab, Kas, Kbs, T
5	A	A, B, S, Kab, Kas, T, Nb, **dec(Nb)**, {Kab, A, T}Kbs
	B	A, B, S, Kab, Kbs, T, Nb, **dec(Nb)**
	S	A, B, S, Kab, Kas, Kbs, T

4.2.4.7 Množina aktuálních předpokladů v průběhu protokolu

1	A	→B: B, S →S: A, B, S, Kas
	B	→S: B, S, Kbs
	S	→A: A, B, S, Kas →B: B, S, Kbs
2	A	→B: B, S →S: A, B, S, **Kab**, Kas, **T**, **{Kab, A, T}Kbs**
	B	→S: B, S, Kbs
	S	→A: A, B, S, **Kab**, Kas, **T**, **{Kab, A, T}Kbs** →B: B, S, Kbs
3	A	→B: A, B, S, **{Kab, A, T}Kbs** →S: A, B, S, Kab, Kas, T, {Kab, A, T}Kbs
	B	→A: **A**, **B**, **{Kab, A, T}Kbs** →S: B, S, Kbs
	S	→A: A, B, S, Kab, Kas, T, {Kab, A, T}Kbs →B: B, S, Kbs

4	A	→B: A, B, S, **{Nb}Kab**, {Kab, A, T}Kbs →S: A, B, S, Kab, Kas, T, {Kab, A, T}Kbs
	B	→A: A, B, **{Nb}Kab**, {Kab, A, T}Kbs →S: B, S, Kbs
	S	→A: A, B, S, Kab, Kas, T, {Kab, A, T}Kbs →B: B, S, Kbs
5	A	→B: A, B, S, Nb, **{dec(Nb)}Kab**, {Kab, A, T}Kbs →S: A, B, S, Kab, Kas, T, {Kab, A, T}Kbs
	B	→A: A, B, Nb, **{dec(Nb)}Kab**, {Kab, A, T}Kbs →S: B, S, Kbs
	S	→A: A, B, S, Kab, Kas, T, {Kab, A, T}Kbs →B: B, S, Kbs

4.2.4.8 Autor

Gavin Lowe 1997

4.2.4.9 Publikace

Reference na protokol: [9].

4.2.1 Kao Chow Authentication v.1

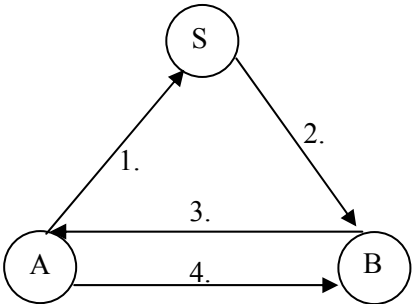
4.2.1.1 Vlastnosti

Protokol Kao Chow Authentication v.1 [10] je protokol, který byl navržen, aby zabránil útokům typu „freshness attak“, který byl úspěšný na opakovanou autentizaci protokolu Neumann Stubblebine. A to přidáním nonce Na do zašifrovaných zpráv v druhém bodě komunikace. Mimo distribuci symetrického klíče, zaručuje autentizaci ve čtvrté zprávě.

4.2.1.2 Specifikace

```
A, B, S:        Subjekty
Na, Nb:         Nonces
Kas, Kbs, Kab:  Klíče

1. A->S: A, B, Na
2. S->B: {A, B, Na, Kab}Kas, {A, B, Na, Kab}Kbs
3. B->A: {A, B, Na, Kab}Kas, {Na}Kab, Nb
4. A->B: {Nb}Kab
```



Obr. 12: Schéma komunikace - Kao Chow Authentication v.1

A – klient
B – klient (klient A chce komunikovat s klientem B)
S – server (server umožní komunikaci. Předá klientovi klíč pro komunikaci)
Kas – klíč pro komunikaci mezi klientem A a serverem
Kbs – klíč pro komunikaci mezi klientem B a serverem
Kab – klíč pro komunikaci mezi klientem A a klientem B

4.2.1.3 Množina počátečních znalostí subjektů

A	A, B, S, Kas
B	B, S, Kbs
S	A, B, S, Kas, Kbs

4.2.1.4 Množina počátečních předpokladů subjektů

A	→B: B, S, Kbs →S: A, B, S, Kas
B	→S: B, S, Kbs
S	→A: A, S, Kas →B: B, S, Kbs

4.2.1.5 Množina cílových znalostí

A	Kab
B	Kab, Nb

4.2.1.6 Množina aktuálních znalostí v průběhu protokolu

1	A	A, B, S, Kas, **Na**
	B	B, S, Kbs
	S	A, B, S, Kas, Kbs, **Na**
2	A	A, B, S, Kas, Na
	B	**A**, B, S, **Kab**, Kbs, **Na**, **{A, B, Na, Kab}Kas**
	S	A, B, S, **Kab**, Kas, Kbs, Na
3	A	A, B, S, **Kab**, Kas, Na, **Nb**
	B	A, B, S, Kab, Kbs, Na, **Nb**, {A, B, Na, Kab}Kas
	S	A, B, S, Kab, Kas, Kbs, Na
4	A	A, B, S, Kab, Kas, Na, Nb
	B	A, B, S, Kab, Kbs, Na, Nb, {A, B, Na, Kab}Kas
	S	A, B, S, Kab, Kas, Kbs, Na

4.2.1.7 Množina aktuálních předpokladů v průběhu protokolu

1	A	→B: B, S, Kbs →S: A, B, S, Kas, **Na**
	B	→S: B, S, Kbs
	S	→A: A, B, S, Kas, **Na** →B: B, S, Kbs
2	A	→B: B, S, Kbs →S: A, B, S, Kas, Na
	B	→A: A, B, S →S: B, S, **Kab**, Kbs, Na, **{A, B, Na, Kab}Kas**
	S	→A: A, B, S, Kas, Na →B: B, S, **Kab**, Kbs, Na, **{A, B, Na, Kab}Kas**
3	A	→B: **A**, B, S, Kbs, Na, **Nb**, **{A, B, Na, Kab}Kas**, **{Na}Kab** →S: A, B, S, Kas, Na
	B	→A: **A**, B, S, Na, **Nb**, **{A, B, Na, Kab}Kas**, **{Na}Kab** →S: B, S, Kab, Kbs, Na, {A, B, Na, Kab}Kas
	S	→A: A, B, S, Kas, Na →B: B, S, Kab, Kbs, Na, {A, B, Na, Kab}Kas

4	A	→B: B, S, Kbs, Na, Nb, {A, B, Na, Kab}Kas, {Na}Kab, **{Nb}Kab** →S: A, B, S, Kas, Na
	B	→A: A, B, S, Na, Nb, {A, B, Na, Kab}Kas, {Na}Kab, **{Nb}Kab** →S: B, S, Kab, Kbs, Na, {A, B, Na, Kab}Kas
	S	→A: A, B, S, Kas, Na →B: B, S, Kab, Kbs, Na, {A, B, Na, Kab}Kas

4.2.1.8 Nedostatky

Tento protokol byl sice navržen, aby odolal útoku „freshness attak“, avšak imunní není. To až jeho verze 2. Útočník odposlouchává zprávy mezi serverem a klienty, a tím je klíč `Kab` kompromitován. V další relaci pak útočník předstírá identitu a může přelstít autentizační část protokolu v bodech 3 a 4 [10].

```
 i.1   A  -> S  :     A, B, Na
 i.2   S  -> B  :     {A, B, Na, Kab}Kas, {A, B, Na, Kab}Kbs
ii.2   Is -> B  :     {A, B, Na, Kab}Kas, {A, B, Na, Kab}Kbs
ii.3   B  -> Ia :     {A, B, Na, Kab}Kas, {Na}Kab, N'b
ii.4   Ia -> B  :     {N'b}Kab
```

4.2.1.9 Autor

I Long Kao a Randy Chow 1995

4.2.1.10 Publikace

Protokol byl uveden v knize zároveň s úspěšným útokem : [10].

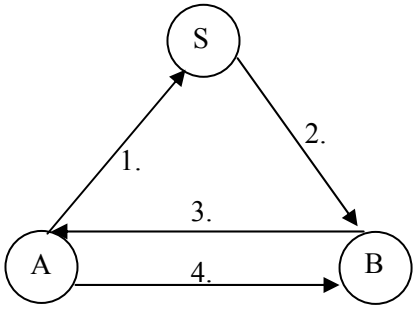
4.2.2 Kao Chow Authentication v.2

4.2.2.1 Vlastnosti

Kao Chow Authentication v.2 [10] je opravený protokol předchozí verze. Verze dva je již odolná proti útoku „freshness attak“, a to přidáním dalšího symetrického klíče `Kt`.

4.2.2.2 Specifikace

A, B, S :	Subjekty
Na, Nb :	Nonces
Kas, Kbs, Kab, Kt :	Klíče



Obr. 13: Schéma komunikace - Kao Chow Authentication v.2

```
1. A->S: A, B, Na
2. S->B: {A, B, Na, Kab, Kt}Kas, {A, B, Na, Kab, Kt}Kbs
3. B->A: B, {A, B, Na, Kab, Kt}Kas, {Na, Kab}Kt, Nb
4. A->B: {Nb, Kab}Kt
```

A – klient
B – klient (klient A chce komunikovat s klientem B)
S – server (server umožní komunikaci. Předá klientovi klíč pro komunikaci)
Kas – klíč pro komunikaci mezi klientem A a serverem
Kbs – klíč pro komunikaci mezi klientem B a serverem
Kab – klíč pro komunikaci mezi klientem A a klientem B
Kt – klíč, který brání proti útokům

4.2.2.3 Množina počátečních znalostí subjektů

A	A, B, S, Kas
B	B, S, Kbs
S	A, B, S, Kas, Kbs

4.2.2.4 Množina počátečních předpokladů subjektů

A	→B: B, S, Kbs →S: A, B, S, Kas
B	→S: B, S, Kbs
S	→A: A, S, Kas →B: B, S, Kbs

4.2.2.5 Množina cílových znalostí

A	Kab
B	Kab, Nb

4.2.2.6 Množina aktuálních znalostí v průběhu protokolu

1	A	A, B, S, Kas, **Na**
	B	B, S, Kbs
	S	A, B, S, Kas, Kbs, **Na**
2	A	A, B, S, Kas, Na
	B	**A**, B, S, **Kab**, Kbs, **Kt**, **Na**, **{A, B, Na, Kab, Kt}Kas**
	S	A, B, S, **Kab**, Kas, Kbs, **Kt**, Na
3	A	A, B, S, **Kab**, Kas, **Kt**, Na, **Nb**
	B	A, B, S, Kab, Kbs, Kt, Na, **Nb**, {A, B, Na, Kab, Kt}Kas
	S	A, B, S, Kab, Kas, Kbs, Kt, Na
4	A	A, B, S, Kab, Kas, Kt, Na, Nb
	B	A, B, S, Kab, Kbs, Kt, Na, Nb, {A, B, Na, Kab, Kt}Kas
	S	A, B, S, Kab, Kas, Kbs, Kt, Na

4.2.2.7 Množina aktuálních předpokladů v průběhu protokolu

1	A	→B: B, S, Kbs →S: A, B, S, Kas, **Na**
	B	→S: B, S, Kbs
	S	→A: A, B, S, Kas, **Na** →B: B, S, Kbs

2	A	→B: B, S, Kbs →S: A, B, S, Kas, Na
	B	→A: A, B, S →S: B, S, **Kab**, Kbs, **Kt**, **Na**, **{A, B, Na, Kab, Kt}Kas**
	S	→A: A, B, S, Kas, Na →B: B, S, Kab, **Kbs**, **Kt**, **Na**, **{A, B, Na, Kab, Kt}Kas**
3	A	→B: **A**, B, S, Kbs, **Na**, **Nb**, **{A, B, Na, Kab, Kt}Kas**, **{Na, Kab}Kt** →S: A, B, S, Kas, Na
	B	→A: **A**, B, S, **Na**, **Nb**, **{A, B, Na, Kab, Kt}Kas**, **{Na, Kab}Kt** →S: B, S, Kab, Kbs, Na, {A, B, Na, Kab, Kt}Kas
	S	→A: A, B, S, Kas, Na →B: B, S, Kab, Kbs, Na, {A, B, Na, Kab, Kt}Kas
4	A	→B: B, S, Kbs, Na, Nb, {A, B, Na, Kab, Kt}Kas, {Na, Kab}Kt, **{Nb, Kab}Kt** →S: A, B, S, Kas, Na
	B	→A: A, B, S, Na, Nb, {A, B, Na, Kab, Kt}Kas, {Na, Kab}Kt, **{Nb, Kab}Kt** →S: B, S, Kab, Kbs, Na, {A, B, Na, Kab, Kt}Kas
	S	→A: A, B, S, Kas, Na →B: B, S, Kab, Kbs, Na, {A, B, Na, Kab, Kt}Kas

4.2.2.8 Autor

I Long Kao a Randy Chow 1995

4.2.2.9 Publikace

Protokol byl uveden v knize: [10].

4.2.3 Wide Mouthed Frog

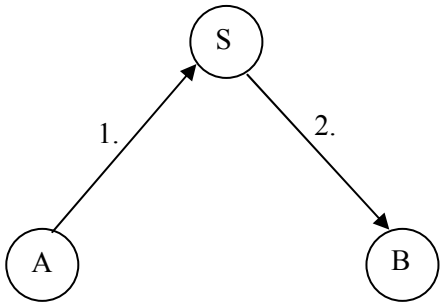
4.2.3.1 Vlastnosti

Protokol Wide Mouthed Frog [11] je velice jednoduchý protokol, který zaručuje distribuci nového symetrického klíče pomocí důvěryhodného serveru. Celá komunikace probíhá pouze ve dvou krocích a zajímavé a odlišné od podobných protokolů je to, že relační klíč nevytváří server, ale klient A. U tohoto protokolu funguje server jako KTC.

4.2.3.2 Specifikace

```
A, B, S :       Subjekty
Ta, Ts          Časová razítka
Kas, Kbs, Kab : Klíče

1.   A->S:   A, {Ta, B, Kab}Kas
2.   S->B:   {Ts, A, Kab}Kbs
```



Obr. 14: Schéma komunikace - Wide Mouthed Frog

A – klient
B – klient (klient A chce komunikovat s klientem B)
S - server (server umožní komunikaci. Předá klientovi A klíč pro komunikaci)
Kas – klíč pro komunikaci mezi klientem A a serverem
Kbs – klíč pro komunikaci mezi klientem B a serverem
Kab – klíč pro komunikaci mezi klientem A a klientem B
Ta – časové razítko vytvořené klientem A
Ts – časové razítko vytvořené serverem S

4.2.3.3 Množina počátečních znalostí subjektů

A	A, B, S, Kas
B	B, S, Kbs
S	A, B, S, Kas, Kbs

4.2.3.4 Množina počátečních předpokladů subjektů

A	→S: A, B, S, Kas →B: B, S
B	→S: B, S, Kbs
S	→A: A, S, Kas →B: B, S, Kbs

4.2.3.5 Množina cílových znalostí

A	Kab
B	Kab

4.2.3.6 Množina aktuálních znalostí v průběhu protokolu

1	A	A, B, S, Kas, **Kab**, **Ta**
	B	B, S, Kbs
	S	A, B, S, Kas, Kbs, **Kab**, **Ta**
2	A	A, B, S, Kas, Kab, Ta
	B	**A**, B, S, Kbs, **Kab**, **Ts**
	S	A, B, S, Kas, Kbs, Kab, Ta, **Ts**

4.2.3.7 Množina aktuálních předpokladů v průběhu protokolu

1	A	→S: A, B, S, Kas, **Kab**, **Ta** →B: B, S
	B	→S: B, S, Kbs
	S	→A: A, S, Kas, **Kab**, **Ta** →B: B, S, Kbs
2	A	→S: A, B, S, Kas, Kab, Ta →B: **A**, B, S
	B	→S: A, B, S, Kbs, **Kab**, **Ts**
	S	→A: A, S, Kas, Kab, Ta →B: A, B, S, Kbs, Kab, **Ts**

4.2.3.8 Nedostatky

Jelikož generování a distribuce klíče je v režii klienta A, útočník může odchytit nové heslo, použít ho pro nově vytvořenou komunikaci s klientem B. Při tomto útoku si klient B myslí, že subjekt A navázal dvě relace avšak klient A o ničem neví [9].

```
  1    A  -> S  :     A, {Ta, B, Kab}Kas
  2    S  -> B  :     {Ts, A, Kab}Kbs
i.2    Is -> B  :     {Ts, A, Kab}Kbs
```

4.2.3.9 Autor

Michael Burrows 1989

4.2.3.10 Publikace

Protokol byl uveden v knize: [11].
Úspěšný útok je popsán v publikaci: [9].

4.2.4 Yahalom

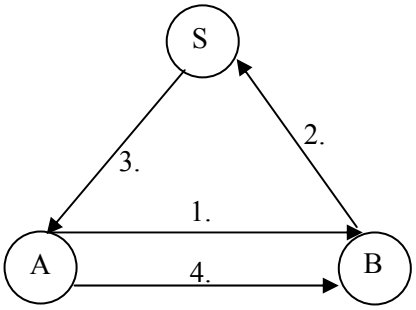
4.2.4.1 Vlastnosti

Protokol Yahalom [11] pracuje na principu trojúhelníkového schématu. Toto schéma vede k možnosti kontroly aktivity jednotlivých subjektů. Šifrování komunikace mezi jednotlivými subjekty je zajištěna pomocí sdíleného symetrického klíče. Symetrický klíč mezi subjekty A a B je generován důvěryhodným serverem. Tento klíč je pak přímo předán subjektu A, který inicializoval komunikaci a následně předán klientovy B právě přes klienta A.

4.2.4.2 Specifikace

```
A, B, S :        Subjekty
Na, Nb :         Nonces
Kas, Kbs, Kab :  Klíče

1. A->B: A, Na
2. B->S: B, {A, Na, Nb}Kbs
3. S->A: {B, Kab, Na, Nb}Kas, {A, Kab}Kbs
4. A->B: {A, Kab}Kbs, {Nb}Kab
```



Obr. 15: Schéma komunikace - Yahalom

A – klient
B – klient (klient A chce komunikovat s klientem B)
S – server (server umožní komunikaci. Předá klientovi klíč pro komunikaci)
Kas – klíč pro komunikaci mezi klientem A a serverem
Kbs – klíč pro komunikaci mezi klientem B a serverem
Kab – klíč pro komunikaci mezi klientem A a klientem B

4.2.4.3 Množina počátečních znalostí subjektů

A	A, B, S, Kas
B	B, S, Kbs
S	A, B, S, Kas, Kbs

4.2.4.4 Množina počátečních předpokladů subjektů

A	→B: B, S, Kbs →S: A, B, S, Kas
B	→S: B, S, Kbs
S	→A: A, B, S, Kas →B: B, S, Kbs

4.2.4.5 Množina cílových znalostí

A	Kab
B	Kab, Nb

4.2.4.6 Množina aktuálních znalostí v průběhu protokolu

1	A	A, B, S, Kas, **Na**
	B	**A**, B, S, Kbs, **Na**
	S	A, B, S, Kas, Kbs
2	A	A, B, S, Kas, Na
	B	A, B, S, Kbs, Na, **Nb**
	S	A, B, S, Kas, Kbs, **Na**, **Nb**
3	A	A, B, S, Kas, **Kab**, Na, **{A, Kab}Kbs**
	B	A, B, S, Kbs, Na, Nb
	S	A, B, S, Kas, Kbs, **Kab**, Na, Nb
4	A	A, B, S, Kas, Kab, Na, {A, Kab}Kbs
	B	A, B, S, Kbs, **Kab**, Na, Nb
	S	A, B, S, Kas, Kbs, Kab, Na, Nb

4.2.4.7 Množina aktuálních předpokladů v průběhu protokolu

1	A	→B: **A**, B, S, Kbs, **Na** →S: A, B, S, Kas
	B	→A: **A**, **B**, **Na** →S: B, S, Kbs
	S	→A: A, B, S, Kas →B: B, S, Kbs
2	A	→B: A, B, S, Kbs, Na →S: A, B, S, Kas
	B	→A: A, B, Na →S: **A**, B, S, Kbs, **Na**, **Nb**
	S	→A: A, B, S, Kas →B: **A**, B, S, Kbs, **Na**, **Nb**
3	A	→B: A, B, S, Kbs, Na →S: A, B, S, Kas, **Kab**, **Na**, **Nb**, **{A, Kab}Kbs**
	B	→A: A, B, Na →S: A, B, S, Kbs, Na, Nb
	S	→A: A, B, S, Kas, **Kab**, **Na**, **Nb**, **{A, Kab}Kbs** →B: A, B, S, Kbs, Na, Nb

4	A	→B: A, B, S, Kbs, Na, **{A, Kab}Kbs, {Nb}Kab** →S: A, B, S, Kas, Kab, Na, Nb, {A, Kab}Kbs
	B	→A: A, B, Na, **{A, Kab}Kbs, {Nb}Kab** →S: A, B, S, Kbs, Na, Nb
	S	→A: A, B, S, Kas, Kab, Na, Nb, {A, Kab}Kbs →B: A, B, S, Kbs, Na, Nb

4.2.4.8 Nedostatky

Slabým místem protokolu Yahalom je předání klíče Kab mezi klientem A a B. Klíč je sice zašifrován pomocí symetrického klíče Kbs, ale do zprávy nejsou přidány žádné nonces. Proto pokud útočník I nějakým způsobem dřív získal neaktuální zprávu, která obsahovala klíč Kab, který je již neaktuální, může beztrestně komunikovat s klientem B, aniž by měl k tomu oprávnění [12].

```
i.1   Ia -> B  :     A, Na
i.2   B  -> Is :     B, Nb, {A, Ni}Kbs
i.4   Ia -> B  :     {A, Kab}Kb, {Ni}Kab
```

4.2.4.9 Autor

Yahalom 1988

4.2.4.10 Publikace

Tento protokol byl publikován v knize: [11].

Útok na protokol Yahalom byl uveden v publikaci: [12].

4.2.5 BAN simplified version of Yahalom

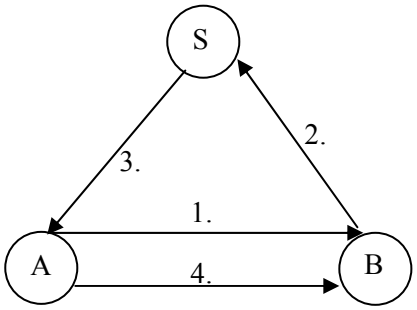
4.2.5.1 Vlastnosti

BAN simplified version of Yahalom [11] je přepracovaná verze protokolu Yahalom. Stále se jedná o protokol, kterým subjekty komunikují v trojúhelníkovém schématu a k distribuci symetrického klíče používají důvěryhodnou třetí stranu. Oproti původnímu protokolu se v třetí zprávě, v části kdy server sděluje nový klíč relace, objevuje nonce Nb. To zamezuje zneužití tohoto protokolu, jak bylo popsáno v předchozí verzi tohoto protokolu.

4.2.5.2 Specifikace

```
A, B, S :        Subjekty
Na, Nb :         Nonces
Kas, Kbs, Kab :  Klíče
```

```
1. A->B: A, Na
2. B->S: B, Nb, {A, Na}Kbs
3. S->A: Nb, {B, Kab, Na}Kas, {A, Kab, Nb}Kbs
4. A->B: {A, Kab, Nb}Kbs, {Nb}Kab
```



Obr. 16: Schéma komunikace - BAN simplified version of Yahalom

A – klient
B – klient (klient A chce komunikovat s klientem B)
S – server (server umožní komunikaci. Předá klientovi klíč pro komunikaci)
Kas – klíč pro komunikaci mezi klientem A a serverem
Kbs – klíč pro komunikaci mezi klientem B a serverem
Kab – klíč pro komunikaci mezi klientem A a klientem B

4.2.5.3 Množina počátečních znalostí subjektů

A	A, B, S, Kas
B	B, S, Kbs
S	A, B, S, Kas, Kbs

4.2.5.4 Množina počátečních předpokladů subjektů

A	→B: B, S, Kbs →S: A, B, S, Kas
B	→S: B, S, Kbs
S	→A: A, B, S, Kas →B: B, S, Kbs

4.2.5.5 Množina cílových znalostí

A	Kab
B	Kab, Nb

4.2.5.6 Množina aktuálních znalostí v průběhu protokolu

1	A	A, B, S, Kas, **Na**
	B	**A**, B, S, Kbs, **Na**
	S	A, B, S, Kas, Kbs
2	A	A, B, S, Kas, Na
	B	A, B, S, Kbs, Na, **Nb**
	S	A, B, S, Kas, Kbs, **Na**, **Nb**
3	A	A, B, S, **Kab**, Kas, Na, **Nb, {A, Kab, Nb}Kbs**
	B	A, B, S, Kbs, Na, Nb
	S	A, B, S, **Kab**, Kas, Kbs, Na, Nb
4	A	A, B, S, Kab, Kas, Na, Nb, {A, Kab, Nb}Kbs
	B	A, B, S, **Kab**, Kbs, Na, Nb
	S	A, B, S, Kab, Kas, Kbs, Na, Nb

4.2.5.7 Množina aktuálních předpokladů v průběhu protokolu

1	A	→B: **A**, B, S, Kbs, **Na** →S: A, B, S, Kas
	B	→A: **A**, **B**, **Na** →S: B, S, Kbs
	S	→A: A, B, S, Kas →B: B, S, Kbs
2	A	→B: A, B, S, Kbs, Na →S: A, B, S, Kas
	B	→A: A, B, Na →S: **A**, B, S, Kbs, **Na**, **Nb**
	S	→A: A, B, S, Kas →B: **A**, B, S, Kbs, **Na**, **Nb**

3	A	→B: A, B, S, Kbs, Na →S: A, B, S, **Kab**, Kas, **Na**, **Nb**, **{A, Kab, Nb}Kbs**
	B	→A: A, B, Na →S: A, B, S, Kbs, Na, Nb
	S	→A: A, B, S, **Kab**, Kas, **Na**, **Nb**, **{A, Kab, Nb}Kbs** →B: A, B, S, Kbs, Na, Nb
4	A	→B: A, B, S, Kbs, Na, **Nb**, **{A, Kab, Nb}Kbs, {Nb}Kab** →S: A, B, S, Kab, Kas, Na, Nb, {A, Kab, Nb}Kbs
	B	→A: A, B, **Na**, **Nb**, **{A, Kab, Nb}Kbs, {Nb}Kab** →S: A, B, S, Kbs, Na, Nb
	S	→A: A, B, S, Kab, Kas, Na, Nb, {A, Kab, Nb}Kbs →B: A, B, S, Kbs, Na, Nb

4.2.5.8 Nedostatky

Přestože je ve zprávě tři přidán nonce Nb, je tento protokol lehce napadnutelný. Útočník využije dvě relace komunikace. V první fázi získá obě nonce Na a Nb. V druhé relaci útočník přesvědčí klienta B, že nonce Na je dvojce (Na, Nb) a získá od něj šifrovanou zprávu {A, Na, Nb}Kbs, kterou pak použije v prvním sezení jako zprávu o novém symetrickém klíči. Proto si bude klient B myslet, že novým klíčem je nonce Na [13].

```
 i.1   A  -> Ib :     A, Na
 i.2   B  -> Is :     B, Nb, {A, Na}Kbs
ii.1   Ia -> B  :     A, Na, Nb
ii.2   B  -> Is :     B, Nb, {A, Na, Nb}Kbs
 i.4   Ia -> B  :     {A, Na, Nb}Kbs, {Nb}Na
```

4.2.5.9 Autor

Burrows Abadi Needham 1989

4.2.5.10 Publikace

Protokol byl uveden v knize: [11].

Útoky na tento protokol byly publikovaný: [13].

4.2.6 Lowe modified version of Yahalom

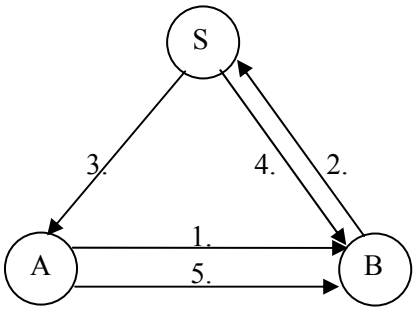
4.2.6.1 Vlastnosti

Další modifikací protokolu Yahalom je Lowe modified version of Yahalom [14], který vytvořil Gavin Lowe v roce 1998. Tato modifikace porušila dřívější schéma trojúhelníku, jelikož symetrický klíč získají klienti přímo od důvěryhodného serveru a nonce `Nb` v druhé zprávě je již zašifrována. Další změnou je také autentizace, která se provádí v přidaném pátém bodu komunikace.

4.2.6.2 Specifikace

```
A, B, S :        Subjekty
Na, Nb :         Nonces
Kas, Kbs, Kab :  Klíče

1.  A->B:  A, Na
2.  B->S:  {A, Na, Nb}Kbs
3.  S->A:  {B, Kab, Na, Nb}Kas
4.  S->B:  {A, Kab}Kbs
5.  A->B:  {A, B, S, Nb}Kab
```



Obr. 17: Schéma komunikace - Lowe modified version of Yahalom

`A` – klient
`B` – klient (klient `A` chce komunikovat s klientem `B`)
`S` – server (server umožní komunikaci. Předá klientovi klíč pro komunikaci)
`Kas` – klíč pro komunikaci mezi klientem `A` a serverem
`Kbs` – klíč pro komunikaci mezi klientem `B` a serverem
`Kab` – klíč pro komunikaci mezi klientem `A` a klientem `B`

4.2.6.3 Množina počátečních znalostí subjektů

A	A, B, S, Kas
B	B, S, Kbs
S	A, B, S, Kas, Kbs

4.2.6.4 Množina počátečních předpokladů subjektů

A	→B: B, S, Kbs →S: A, B, S, Kas
B	→S: B, S, Kbs
S	→A: A, B, S, Kas →B: B, S, Kbs

4.2.6.5 Množina cílových znalostí

A	Kab
B	Kab, Nb

4.2.6.6 Množina aktuálních znalostí v průběhu protokolu

1	A	A, B, S, Kas, **Na**
	B	**A**, B, S, Kbs, **Na**
	S	A, B, S, Kas, Kbs
2	A	A, B, S, Kas, Na
	B	A, B, S, Kbs, Na, **Nb**
	S	A, B, S, Kas, Kbs, **Na**, **Nb**
3	A	A, B, S, **Kab**, Kas, **Na**, **Nb**
	B	A, B, S, Kbs, Na, Nb
	S	A, B, S, **Kab**, Kas, Kbs, Na, Nb
4	A	A, B, S, Kab, Kas, Na, Nb
	B	A, B, S, **Kab**, Kbs, Na, Nb
	S	A, B, S, Kab, Kas, Kbs, Na, Nb
5	A	A, B, S, Kab, Kas, Na, Nb
	B	A, B, S, Kab, Kbs, Na, Nb
	S	A, B, S, Kab, Kas, Kbs, Na, Nb

4.2.6.7 Množina aktuálních předpokladů v průběhu protokolu

1	A	→B: **A**, B, S, Kbs, **Na** →S: A, B, S, Kas
	B	→A: **A**, B, **Na** →S: B, S, Kbs
	S	→A: A, B, S, Kas →B: B, S, Kbs
2	A	→B: A, B, S, Kbs, Na →S: A, B, S, Kas
	B	→A: A, B, Na →S: **A**, B, S, Kbs, **Na**, **Nb**
	S	→A: A, B, S, Kas →B: **A**, B, S, Kbs, **Na**, **Nb**
3	A	→B: A, B, S, Kbs, Na →S: A, B, S, **Kab**, Kas, **Na**, **Nb**
	B	→A: A, B, Na →S: A, B, S, Kbs, Na, Nb
	S	→A: A, B, S, **Kab**, Kas, **Na**, **Nb** →B: A, B, S, Kbs, Na, Nb
4	A	→B: A, B, S, Kbs, Na →S: A, B, S, Kab, Kas, Na, Nb
	B	→A: A, B, Na →S: A, B, S, **Kab**, Kbs, Na, Nb
	S	→A: A, B, S, Kab, Kas, Na, Nb →B: A, B, S, **Kab**, Kbs, Na, Nb
5	A	→B: A, B, S, Kbs, Na, Nb, **{A, B, S, Nb}Kab** →S: A, B, S, Kab, Kas, Na, Nb
	B	→A: A, B, Na, Nb, **{A, B, S, Nb}Kab** →S: A, B, S, Kab, Kbs, Na, Nb
	S	→A: A, B, S, Kab, Kas, Na, Nb →B: A, B, S, Kab, Kbs, Na, Nb

4.2.6.8 Autor

Gavin Lowe 1998

4.2.6.9 Publikace

Protokol byl uveden v článku: [14].

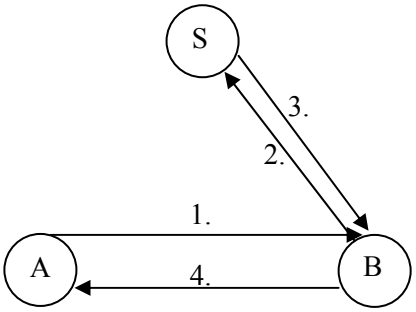
4.2.7 Otway Rees

4.2.7.1 Vlastnosti

Protokol Otway Rees [15] je určen pro distribuci symetrického klíče mezi dvěma klienty, kteří chtějí komunikovat v nezabezpečené síti. Zajímavostí tohoto protokolu je nonce `M`, který je přidán do každé zprávy a má funkci čísla jednotlivých relací.

4.2.7.2 Specifikace

```
A, B, S :          Subjekty
M, Na, Nb :        Nonces
Kab, Kas, Kbs :    Klíče
```



Obr. 18: Schéma komunikace - Otway Rees

```
1.  A->B :  M, A, B, {Na, M, A, B}Kas
2.  B->S :  M, A, B, {Na, M, A, B}Kas , {Nb, M, A, B}Kbs
3.  S->B :  M, {Na, Kab}Kas, {Nb, Kab}Kbs
4.  B->A :  M, {Na, Kab}Kas
```

`A` – klient
`B` – klient (klient `A` chce komunikovat s klientem `B`)
`S` – server, distribuce klíčů
`Kab` – klíč pro komunikaci mezi klientem `A` a `B`
`Kas` – klíč pro komunikaci mezi klientem `A` a serverem `S`
`Kbs` – klíč pro komunikaci mezi klientem `B` a serverem `S`
`M` – nonce, číslo relace

4.2.7.3 Množina počátečních znalostí subjektů

A	A, B, S, Kas
B	B, S, Kbs
S	A, B, S, Kas, Kbs

4.2.7.4 Množina počátečních předpokladů subjektů

A	→B: B, S →S: A, B, S, Kas
B	→S: B, S, Kbs
S	→A: A, S, Kas →B: B, S, Kbs

4.2.7.5 Množina cílových znalostí subjektů

A	Kab, Na
B	Kab

4.2.7.6 Množina aktuálních znalostí v průběhu protokolu

1	A	A, B, S, Kas, **M, Na**
	B	**A**, B, S, Kbs, **M, {Na, M, A, B}Kas**
	S	A, B, S, Kas, Kbs
2	A	A, B, S, Kas, M, Na
	B	A, B, S, Kbs, M, **Nb**, {Na, M, A, B}Kas
	S	A, B, S, Kas, Kbs, **M, Na, Nb**
3	A	A, B, S, Kas, M, Na
	B	A, B, S, **Kab**, Kbs, M, Nb, {Na, M, A, B}Kas, **{Na, Kab}Kas**
	S	A, B, S, **Kab**, Kas, Kbs, M, Na, Nb
4	A	A, B, S, **Kab**, Kas, M, Na
	B	A, B, S, Kab, Kbs, M, Nb, {Na, M, A, B}Kas, {Na, Kab}Kas
	S	A, B, S, Kab, Kas, Kbs, M, Na, Nb

4.2.7.7 Množina aktuálních předpokladů v průběhu protokolu

1	A	→B: **A**, B, S, **M, {Na, M, A, B}Kas** →S: A, B, S, Kas
	B	→A: **A**, B, **M, {Na, M, A, B}Kas** →S: B, S, Kbs
	S	→A: A, S, Kas →B: B, S, Kbs
2	A	→B: A, B, S, M, {Na, M, A, B}Kas →S: A, B, S, Kas
	B	→A: A, B, M, {Na, M, A, B}Kas →S: **A**, B, S, Kbs, **M, Nb, {Na, M, A, B}Kas**
	S	→A: A, S, Kas, M, Na →B: **A**, B, S, Kbs, **M, Nb, {Na, M, A, B}Kas**
3	A	→B: A, B, S, M, {Na, M, A, B}Kas →S: A, B, S, Kas
	B	→A: A, B, M, {Na, M, A, B}Kas →S: A, B, S, **Kab**, Kbs, M, Nb, {Na, M, A, B}Kas, **{Na, Kab}Kas**
	S	→A: A, S, Kas, M, Na →B: A, B, S, **Kab**, Kbs, M, Nb, {Na, M, A, B}Kas, **{Na, Kab}Kas**
4	A	→B: A, B, S, M, {Na, M, A, B}Kas, **{Na, Kab}Kas** →S: A, B, S, Kas
	B	→A: A, B, M, {Na, M, A, B}Kas, **{Na, Kab}Kas** →S: A, B, S, Kab, Kbs, M, Nb, {Na, M, A, B}Kas, {Na, Kab}Kas
	S	→A: A, S, Kas, M, Na →B: A, B, S, Kab, Kbs, M, Nb, {Na, M, A, B}Kas

4.2.7.8 Nedostatky

Jelikož nedochází k žádné autentizaci při výměně certifikátů mezi klienty A a B, je protokol Otway rees napadnutelný odposlechem a následně podvržením identity [16]. Pokud útočník I zachytí první zprávu od klienta A a obratem pošle zašifrovaný certifikát {Na, M, A, B}Kas zpět, bude se subjekt A domnívat, že nový symetrický klíč relace je trojce M, A, B, a bude komunikovat s útočníkem pomocí tohoto klamného klíče.

```
i.1   A  -> Ib  :    M, A, B, {Na, M, A, B}Kas
i.2   B  -> S   :    M, A, B, {Na, M, A, B}Kas , {Nb, M, A, B}Kbs
i.3   S  -> B   :    M, {Na, Kab}Kas, {Nb, Kab}Kbs
i.4   Ib -> A   :    M, {Na, M, A, B}Kas
```

4.2.7.9 Autor

D. Otway a O. Rees January 1997

4.2.7.10 Publikace

Reference na protokol: [15].

Útoky na tento protokol: [16].

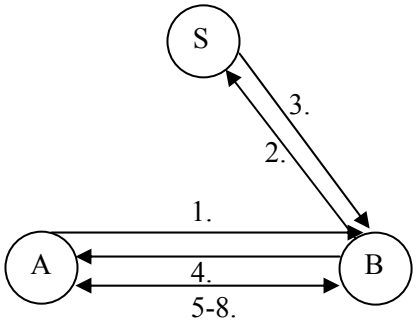
4.2.8 KSL

4.2.8.1 Vlastnosti

Protokol KSL [17] využívá k distribuci symetrického klíče třetí stranu (body 1-5 uvedené níže) a vzájemnou autentizaci si zařizují samotní klienti (6-8). Proto je v protokolu použito více nonces. Klient B vytvoří lístek pro klienta A, který se tímto prostředkem může opakovaně autentizovat, dokud nevyprší platnost časového razítka uvnitř lístku. Proto se této autentizaci říká opakovaná (*repeated authentication*).

4.2.8.2 Specifikace

```
A, B, S :              Subjekty
Na, Nb, Nc, Ma, Mb :   Nonce
Kas, Kbs, Kab, Kbb :   Klíče
Tb :                   Časové razítko
```



Obr. 19: Schéma komunikace - KSL

```
1.  A->B:  Na, A
2.  B->S:  Na, A, Nb, B
3.  S->B:  {Nb, A, Kab}Kbs, {Na, B, Kab}Kas
4.  B->A:  {Na, B, Kab}Kas, {Tb, A, Kab}Kbb, Nc, {Na}Kab
5.  A->B:  {Nc}Kab
6.  A->B:  Ma, {Tb, A, Kab}Kbb
7.  B->A:  Mb, {Ma}Kab
8.  A->B:  {Mb}Kab
```

A – klient
B – klient (klient A chce komunikovat s klientem B)
S – server, distribuce klíčů
Kab – klíč pro komunikaci mezi klientem A a B
Kas – klíč pro komunikaci mezi klientem A a serverem S
Kbs – klíč pro komunikaci mezi klientem B a serverem S
Kbb – klíč, kterým je šifrována zpráva (lístek) pro pozdější autentizaci. Tento klíč zná pouze klient B

4.2.8.3 Množina počátečních znalostí subjektů

A	A, B, S, Kas
B	B, S, Kbs, Kbb
S	A, B, S, Kas, Kbs

4.2.8.4 Množina počátečních předpokladů subjektů

A	→B: B, S →S: A, B, S, Kas
B	→S: B, S, Kbs
S	→A: A, S, Kas →B: B, S, Kbs

4.2.8.5 Množina cílových znalostí subjektů

A	Kab, {Tb, A, Kab}Kbb
B	Kab

4.2.8.6 Množina aktuálních znalostí v průběhu protokolu

1	A	A, B, S, Kas, **Na**
	B	**A**, B, S, Kbs, Kbb, **Na**
	S	A, B, S, Kas, Kbs
2	A	A, B, S, Kas, Na
	B	A, B, S, Kbs, Kbb, Na, **Nb**
	S	A, B, S, Kas, Kbs, **Na**, **Nb**
3	A	A, B, S, Kas, Na
	B	A, B, S, **Kab**, Kbs, Kbb, Na, Nb, **{Na, B, Kab}Kas**
	S	A, B, S, **Kab**, Kas, Kbs, Na, Nb
4	A	A, B, S, **Kab**, Kas, Na, **Nc**, **{Tb, A, Kab}Kbb**
	B	A, B, S, Kab, Kbs, Kbb, Na, Nb, **Nc**, {Na, B, Kab}Kas, **Tb**
	S	A, B, S, Kab, Kas, Kbs, Na, Nb
5	A	A, B, S, Kab, Kas, Na, Nc, {Tb, A, Kab}Kbb
	B	A, B, S, Kab, Kbs, Kbb, Na, Nb, Nc, {Na, B, Kab}Kas, Tb
	S	A, B, S, Kab, Kas, Kbs, Na, Nb
6	A	A, B, S, Kab, Kas, Na, Nc, **Ma**, {Tb, A, Kab}Kbb
	B	A, B, S, Kab, Kbs, Kbb, Na, Nb, Nc, **Ma**, {Na, B, Kab}Kas, Tb
	S	A, B, S, Kab, Kas, Kbs, Na, Nb
7	A	A, B, S, Kab, Kas, Na, Nc, Ma, **Mb**, {Tb, A, Kab}Kbb
	B	A, B, S, Kab, Kbs, Kbb, Na, Nb, Nc, Ma, **Mb**, {Na, B, Kab}Kas, Tb
	S	A, B, S, Kab, Kas, Kbs, Na, Nb
8	A	A, B, S, Kab, Kas, Na, Nc, Ma, Mb, {Tb, A, Kab}Kbb
	B	A, B, S, Kab, Kbs, Kbb, Na, Nb, Nc, Ma, Mb, {Na, B, Kab}Kas, Tb
	S	A, B, S, Kab, Kas, Kbs, Na, Nb

4.2.8.7 Množina aktuálních předpokladů v průběhu protokolu

1	A	→B: **A**, B, S, **Na** →S: A, B, S, Kas
	B	→A: **A**, **B**, **Na** →S: B, S, Kbs
	S	→A: A, S, Kas →B: B, S, Kbs
2	A	→B: A, B, S, Na →S: A, B, S, Kas
	B	→A: A, B, Na →S: **A**, B, S, Kbs, **Na**, **Nb**
	S	→A: A, S, Kas →B: A, B, S, Kbs, **Na**, **Nb**
3	A	→B: A, B, S, Na →S: A, B, S, Kas
	B	→A: A, B, Na →S: A, B, S, Kbs, **Kab**, Na, Nb, **{Na, B, Kab}Kas**
	S	→A: A, S, Kas →B: A, B, S, Kbs, **Kab**, Na, Nb, **{Na, B, Kab}Kas**
4	A	→B: A, B, S, Na, **Nc** {Na, B, Kab}Kas, **{Tb, A, Kab}Kbb, {Na}Kab** →S: A, B, S, Kas
	B	→A: A, B, Na, **Nc**, {Na, B, Kab}Kas, **{Tb, A, Kab}Kbb, {Na}Kab** →S: A, B, S, Na, Nb, Kbs, Kab, {Na, B, Kab}Kas
	S	→A: A, S, Kas →B: A, B, S, Kab, Na, Nb, Kbs, {Na, B, Kab}Kas
5	A	→B: A, B, S, Na, Nc {Na, B, Kab}Kas, {Tb, A, Kab}Kbb, {Na}Kab, **{Nc}Kab** →S: A, B, S, Kas
	B	→A: A, B, Na, Nc, {Na, B, Kab}Kas, {Tb, A, Kab}Kbb, {Na}Kab, **{Nc}Kab** →S: A, B, S, Kbs, Kab, Na, Nb, {Na, B, Kab}Kas
	S	→A: A, S, Kas →B: A, B, S, Kbs, Kab, Na, Nb, {Na, B, Kab}Kas
6	A	→B: A, B, S, Na, Nc, **Ma**, {Na, B, Kab}Kas, {Tb, A, Kab}Kbb, {Na}Kab, {Nc}Kab →S: A, B, S, Kas
	B	→A: A, B, Na, Nc, **Ma**, {Na, B, Kab}Kas, {Tb, A, Kab}Kbb, {Na}Kab, {Nc}Kab →S: A, B, S, Kbs, Kab, Na, Nb, {Na, B, Kab}Kas
	S	→A: A, S, Kas →B: A, B, S, Kbs, Kab, Na, Nb, {Na, B, Kab}Kas
7	A	→B: A, B, S, Na, Nc, Ma, **Mb**, {Na, B, Kab}Kas, {Tb, A, Kab}Kbb, {Na}Kab, {Nc}Kab, **{Ma}Kab** →S: A, B, S, Kas
	B	→A: A, B, Na, Nc, Ma, **Mb**, {Na, B, Kab}Kas, {Tb, A, Kab}Kbb, {Na}Kab, {Nc}Kab, **{Ma}Kab** →S: A, B, S, Kbs, Kab, Na, Nb, {Na, B, Kab}Kas
	S	→A: A, S, Kas →B: A, B, S, Kbs, Kab, Na, Nb, {Na, B, Kab}Kas

8	A	→B: A, B, S, Na, Nc, Ma, Mb, {Na, B, Kab}Kas, {Tb, A, Kab}Kbb, {Na}Kab, {Nc}Kab, {Ma}Kab, **{Mb}Kab** →S: A, B, S, Kas
	B	→A: A, B, Na, Nc, Ma, Mb, {Na, B, Kab}Kas, {Tb, A, Kab}Kbb, {Na}Kab, {Nc}Kab, {Ma}Kab, **{Mb}Kab** →S: A, B, S, Kbs, Kab, Na, Nb, {Na, B, Kab}Kas
	S	→A: A, S, Kas →B: A, B, S, Kbs, Kab, Na, Nb, {Na, B, Kab}Kas

4.2.8.8 Nedostatky

Útočník může druhou část protokolu KLS zneužít pro zašifrování svých dat pomocí klíče `Kab` [18]. K tomu použije odposlechnutý lístek `{Tb, A, Kab}Kbb` ze 4. zprávy. Takže pokud chce útočník `I` zašifrovat data `D` (`{D}Kab`), zneužije opakovanou autentizaci takto:

```
i.6.  Ia -> B :      D, {Tb, A, Kab}Kbb
i.7.  B -> Ia :      Mb, {D}Kab
```

Pokud chce útočník komunikovat s klientem `B` a vydávat se za klienta `A`, může hned využít dvou způsobů. Opět zneužije lístku z předchozích zpráv a s podvrženou identitou komunikuje buď s oběma klienty (musí proběhnout celý protokol, aby útočník získal i lístek zašifrovaný klíčem `Kaa`) [18]

```
 i.6   Ia -> B :      Mi, {Tb, A, Kab}Kbb
 i.7   B  -> Ia :     Mb, {Mi}Kab
ii.6   Ib -> A :      Mb, {Ta, B, Kab}Kaa
ii.7   A  -> Ib :     Ma, {Mb}Kab
 i.8   Ia -> B        {Mb}Kab
```

nebo jen s klientem `B` [19].

```
 i.6   Ia -> B :      Mi, { Tb, A, Kab}Kbb
 i.7   B  -> Ia :     Mb, {Mi}Kab
ii.6   Ia -> B :      Mb, {Tb, A, Kab}Kbb
ii.7   B  -> Ia :     Mb', {Mb}Kab
 i.8   Ia -> B        {Mb}Kab
```

4.2.8.9 Autor

Axel Kehne, Jürgen Schönwälder a Horst Langendörfer 1992

4.2.8.10 Publikace

Reference na protokol: [17].

Šifrování pomocí opakované autentizace a podvržení identity pomocí dvou klientů: [18].

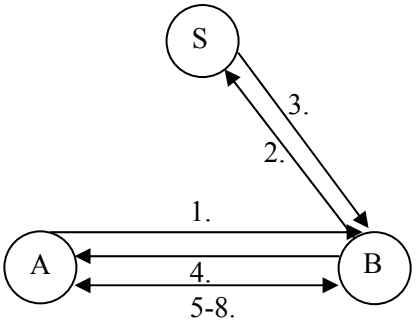
Podvržení identity pomocí jednoho klienta: [20].

4.2.9 Lowe modified KSL

4.2.9.1 Vlastnosti

V roce 1996 Gavin Lowe upravil protokol KLS tak, aby byl schopen čelit autentizačním útokům, které zneužívali lístky, které jsou vytvářeny pro přístup k subjektu B. Tím vznikl protokol Lowe modified KSL [18]. V tomto protokolu v bodě 4, 7 a 8 přibyla manuální identifikace komunikujících subjektů. To zabránilo použití vytvořeného lísku útočníkům jiné identity.

4.2.9.2 Specifikace



Obr. 20: Schéma komunikace - Lowe modified KSL

```
A, B, S :                Subjekty
Na, Nb, Nc, Ma, Mb :     Nonce
Kas, Kbs, Kab, Kbb :     Klíče
Tb :                     Časové razítko

1.  A->B:  Na, A
2.  B->S:  Na, A, Nb, B
3.  S->B:  {Nb, A, Kab}Kbs, {Na, B, Kab}Kas
4.  B->A:  {Na, B, Kab}Kas, {Tb, A, Kab}Kbb, Nc, {B, Na}Kab
5.  A->B:  {Nc}Kab
6.  A->B:  Ma, {Tb, A, Kab}Kbb
7.  B->A:  Mb, {Ma, B}Kab
8.  A->B:  {A, Mb}Kab
```

A – klient
B – klient (klient A chce komunikovat s klientem B)
S – server, distribuce klíčů
Kab – klíč pro komunikaci mezi klientem A a B
Kas – klíč pro komunikaci mezi klientem A a serverem S
Kbs – klíč pro komunikaci mezi klientem B a serverem S
Kbb – klíč, kterým je šifrována zpráva (lístek) pro pozdější autentizaci. Tento klíč zná pouze klient B

4.2.9.3 Množina počátečních znalostí subjektů

A	A, B, S, Kas
B	B, S, Kbs, Kbb
S	A, B, S, Kas, Kbs

4.2.9.4 Množina počátečních předpokladů subjektů

A	→B: B, S →S: A, B, S, Kas
B	→S: B, S, Kbs
S	→A: A, S, Kas →B: B, S, Kbs

4.2.9.5 Množina cílových znalostí subjektů

A	Kab, {Tb, A, Kab}Kbb
B	Kab, Kbb

4.2.9.6 Množina aktuálních znalostí v průběhu protokolu

1	A	A, B, S, Kas, **Na**
	B	**A**, B, S, Kbs, Kbb, **Na**
	S	A, B, S, Kas, Kbs
2	A	A, B, S, Kas, Na
	B	A, B, S, Kbs, Kbb, Na, **Nb**
	S	A, B, S, Kas, Kbs, **Na**, **Nb**
3	A	A, B, S, Kas, Na
	B	A, B, S, **Kab**, Kbs, Kbb, Na, Nb, **{Na, B, Kab}Kas**
	S	A, B, S, **Kab**, Kas, Kbs, Na, Nb
4	A	A, B, S, **Kab**, Kas, Na, **Nc**, **{Tb, A, Kab}Kbb**
	B	A, B, S, Kab, Kbs, Kbb, Na, Nb, **Nc**, {Na, B, Kab}Kas, **Tb**
	S	A, B, S, Kab, Kas, Kbs, Na, Nb
5	A	A, B, S, Kab, Kas, Na, Nc, {Tb, A, Kab}Kbb
	B	A, B, S, Kab, Kbs, Kbb, Na, Nb, Nc, {Na, B, Kab}Kas, Tb
	S	A, B, S, Kab, Kas, Kbs, Na, Nb
6	A	A, B, S, Kab, Kas, Na, Nc, **Ma**, {Tb, A, Kab}Kbb
	B	A, B, S, Kab, Kbs, Kbb, Na, Nb, Nc, **Ma**, {Na, B, Kab}Kas, Tb
	S	A, B, S, Kab, Kas, Kbs, Na, Nb
7	A	A, B, S, Kab, Kas, Na, Nc, Ma, **Mb**, {Tb, A, Kab}Kbb
	B	A, B, S, Kab, Kbs, Kbb, Na, Nb, Nc, Ma, **Mb**, {Na, B, Kab}Kas, Tb
	S	A, B, S, Kab, Kas, Kbs, Na, Nb
8	A	A, B, S, Kab, Kas, Na, Nc, Ma, Mb, {Tb, A, Kab}Kbb
	B	A, B, S, Kab, Kbs, Kbb, Na, Nb, Nc, Ma, Mb, {Na, B, Kab}Kas, Tb
	S	A, B, S, Kab, Kas, Kbs, Na, Nb

4.2.9.7 Množina aktuálních předpokladů v průběhu protokolu

1	A	→B: **A**, B, S, **Na** →S: A, B, S, Kas
	B	→A: **A**, **B**, **Na** →S: B, S, Kbs
	S	→A: A, S, Kas →B: B, S, Kbs
2	A	→B: A, B, S, Na →S: A, B, S, Kas
	B	→A: A, B, Na →S: **A**, B, S, Kbs, **Na**, **Nb**
	S	→A: A, S, Kas →B: A, B, S, Kbs, **Na**, **Nb**
3	A	→B: A, B, S, Na →S: A, B, S, Kas
	B	→A: A, B, Na →S: A, B, S, Kbs, **Kab**, Na, Nb, **{Na, B, Kab}Kas**
	S	→A: A, S, Kas →B: A, B, S, Kbs, **Kab**, Na, Nb, **{Na, B, Kab}Kas**

4	A	→B: A, B, S, Na, **Nc** {Na, B, Kab}Kas, **{Tb, A, Kab}Kbb, {B, Na}Kab** →S: A, B, S, Kas
	B	→A: A, B, Na, **Nc**, {Na, B, Kab}Kas, **{Tb, A, Kab}Kbb, {B, Na}Kab** →S: A, B, S, Na, Nb, Kbs, Kab, {Na, B, Kab}Kas
	S	→A: A, S, Kas →B: A, B, S, Kab, Na, Nb, Kbs, {Na, B, Kab}Kas
5	A	→B: A, B, S, Na, Nc {Na, B, Kab}Kas, {Tb, A, Kab}Kbb, {B, Na}Kab, **{Nc}Kab** →S: A, B, S, Kas
	B	→A: A, B, Na, Nc, {Na, B, Kab}Kas, {Tb, A, Kab}Kbb, {B, Na}Kab, **{Nc}Kab** →S: A, B, S, Kbs, Kab, Na, Nb, {Na, B, Kab}Kas
	S	→A: A, S, Kas →B: A, B, S, Kbs, Kab, Na, Nb, {Na, B, Kab}Kas
6	A	→B: A, B, S, Na, Nc, **Ma**, {Na, B, Kab}Kas, {Tb, A, Kab}Kbb, {B, Na}Kab, {Nc}Kab →S: A, B, S, Kas
	B	→A: A, B, Na, Nc, **Ma**, {Na, B, Kab}Kas, {Tb, A, Kab}Kbb, {B, Na}Kab, {Nc}Kab →S: A, B, S, Kbs, Kab, Na, Nb, {Na, B, Kab}Kas
	S	→A: A, S, Kas →B: A, B, S, Kbs, Kab, Na, Nb, {Na, B, Kab}Kas
7	A	→B: A, B, S, Na, Nc, Ma, **Mb**, {Na, B, Kab}Kas, {Tb, A, Kab}Kbb, {B, Na}Kab, {Nc}Kab, **{Ma, B}Kab** →S: A, B, S, Kas
	B	→A: A, B, Na, Nc, Ma, **Mb**, {Na, B, Kab}Kas, {Tb, A, Kab}Kbb, {B, Na}Kab, {Nc}Kab, **{Ma, B}Kab** →S: A, B, S, Kbs, Kab, Na, Nb, {Na, B, Kab}Kas
	S	→A: A, S, Kas →B: A, B, S, Kbs, Kab, Na, Nb, {Na, B, Kab}Kas
8	A	→B: A, B, S, Na, Nc, Ma, Mb, {Na, B, Kab}Kas, {Tb, A, Kab}Kbb, {B, Na}Kab, {Nc}Kab, {Ma, B}Kab, **{A, Mb}Kab** →S: A, B, S, Kas
	B	→A: A, B, Na, Nc, Ma, Mb, {Na, B, Kab}Kas, {Tb, A, Kab}Kbb, {B, Na}Kab, {Nc}Kab, {Ma, B}Kab, **{A, Mb}Kab** →S: A, B, S, Kbs, Kab, Na, Nb, {Na, B, Kab}Kas
	S	→A: A, S, Kas →B: A, B, S, Kbs, Kab, Na, Nb, {Na, B, Kab}Kas

4.2.9.8 Autor

Gavin Lowe 1996

4.2.9.9 Publikace

Reference na protokol: [18].

4.2.10 Woo and Lam Pi

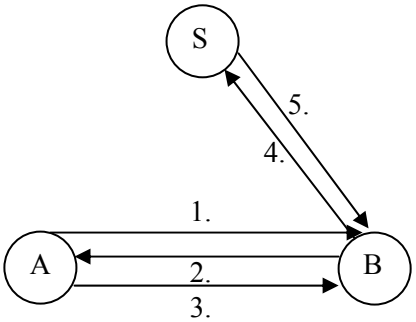
4.2.10.1 Vlastnosti

Protokol Woo And Lam Pi [19] je protokolem čistě autentizačním. Za jeho pomoci si subjekt B ověří identitu klienta A přes důvěryhodný server S.

4.2.10.2 Specifikace

```
A, B, S :     Subjekty
Nb :          Nonce
Kas, Kbs :    Klíč

1.  A->B:  A
2.  B->A:  Nb
3.  A->B:  {Nb}Kas
4.  B->S:  {A, {Nb}Kas}Kas
5.  S->B:  {Nb}Kbs
```



Obr. 21: Schéma komunikace - Woo And Lam Pi

A – klient
B – klient (klient A chce komunikovat s klientem B)
S – server, důvěryhodná třetí strana
Kas – klíč pro komunikaci mezi klientem A a serverem S
Kab – klíč pro komunikaci mezi klientem B a serverem S

4.2.10.3 Množina počátečních znalostí subjektů

A	A, B, S, Kas
B	B, S, Kbs
S	A, B, S, Kas, Kbs

4.2.10.4 Množina počátečních předpokladů subjektů

A	→B: B, S →S: A, B, S, Kas
B	→S: B, S, Kbs
S	→A: A, S, Kas →B: B, S, Kbs

4.2.10.5 Množina cílových znalostí subjektů

A	Nb
B	Nb

4.2.10.6 Množina aktuálních znalostí v průběhu protokolu

1	A	A, B, S, Kas
	B	**A**, B, S, Kbs
	S	A, B, S, Kas, Kbs

2	A	A, B, S, Kas, **Nb**
	B	A, B, S, Kbs, **Nb**
	S	A, B, S, Kas, Kbs
3	A	A, B, S, Kas, Nb
	B	A, B, S, Kbs, Nb, **{Nb}Kas**
	S	A, B, S, Kas, Kbs
4	A	A, B, S, Kas, Nb
	B	A, B, S, Kbs, Nb, {Nb}Kas
	S	A, B, S, Kas, Kbs, **Nb**
5	A	A, B, S, Kas, Nb
	B	A, B, S, Kbs, Nb, {Nb}Kas
	S	A, B, S, Kas, Kbs, Nb

4.2.10.7 Množina aktuálních předpokladů v průběhu protokolu

1	A	→B: **A**, B, S →S: A, B, S, Kas
	B	→A: **A**, **B** →S: B, S, Kbs
	S	→A: A, S, Kas →B: B, S, Kbs
2	A	→B: A, B, S, **Nb** →S: A, B, S, Kas
	B	→A: A, B, **Nb** →S: B, S, Kbs
	S	→A: A, S, Kas →B: B, S, Kbs
3	A	→B: A, B, S, Nb, **{Nb}Kas** →S: A, B, S, Kas
	B	→A: A, B, Nb, **{Nb}Kas** →S: B, S, Kbs
	S	→A: A, S, Kas →B: B, S, Kbs
4	A	→B: A, B, S, Nb, {Nb}Kas →S: A, B, S, Kas
	B	→A: A, B, Nb, {Nb}Kas →S: B, S, Kbs, **{Nb}Kas**
	S	→A: A, S, Kas →B: **A**, B, S, Kbs, **{Nb}Kas**
5	A	→B: A, B, S, Nb, {Nb}Kas →S: A, B, S, Kas
	B	→A: A, B, Nb, {Nb}Kas →S: B, S, Kbs, **Nb**, {Nb}Kas
	S	→A: A, S, Kas →B: A, B, S, Kbs, **Nb**, {Nb}Kas

4.2.10.8 Autor

Woo a Lam 1994

4.2.10.9 Publikace

Reference na protokol: [19].

4.3 Protokoly s více subjekty a s asymetrickým šifrováním

Do této skupiny protokolů jsem zařadil protokoly, které k zašifrování zpráv mezi klienty a důvěryhodným serverem používají asymetrické šifrování. Tedy dvojice soukromých a veřejných klíčů. Zpracoval jsem tyto protokoly: Needham-Schroeder Public Key, Lowe's fixed version of Needdaham-Schroeder public Key a TMN.

4.3.1 Needham-Schroeder Public Key

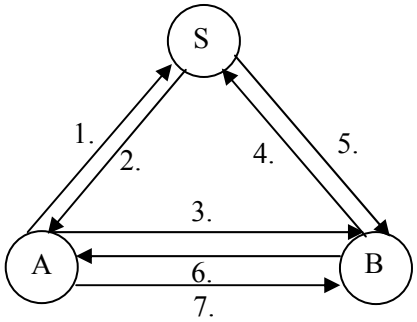
4.3.1.1 Vlastnosti

Needham-Schroeder Public Key [6] je protokol od autorů R. Needham a M. Schroeder, který pro chráněnou komunikaci mezi subjekty používá asymetrické šifrování. K tomu slouží páry veřejných a soukromých klíčů pro všechny zúčastněné subjekty. Jednotlivé veřejné klíče klienti získají od třetí strany, důvěryhodného serveru. Protokol slouží k oboustranné autentizaci.

4.3.1.2 Specifikace

```
A, B, S :                    Subjekty
Na, Nb :                     Nonce
PKa,PKb,PKs,SKa,SKb,SKs :    Veřejné a soukromé klíče

1.  A->S:  A, B
2.  S->A:  {PKb, B}SKs
3.  A->B:  {Na, A}PKb
4.  B->S:  B, A
5.  S->B:  {PKa, A}SKs
6.  B->A:  {Na, Nb}PKa
7.  A->B:  {Nb}PKb
```



Obr. 22: Schéma komunikace - Needham-Schroeder Public Key

A – klient
B – klient (klient A chce komunikovat s klientem B)
S – server (server umožní komunikaci. Předá klientovi a klíč pro komunikaci)
PKa, SKa – veřejný a soukromí klíč klienta A
PKb, SKb – veřejný a soukromí klíč klienta B
PKs, SKs – veřejný a soukromí klíč serveru S

4.3.1.3 Množina počátečních znalostí subjektů

A	A, B, S, SKa, PKs
B	B, S, SKb, PKs
S	A, B, S, SKs, PKa, PKb

4.3.1.4 Množina počátečních předpokladů subjektů

A	→B: B, S, PKs, SKb →S: A, B, S, PKa, SKs
B	→S: B, S, PKb, SKs
S	→A: A, S, PKs, SKa →B: B, S, PKs, SKb

4.3.1.5 Množina cílových znalostí

A	PKb, Na
B	PKa, Nb

4.3.1.6 Množina aktuálních znalostí

1	A	A, B, S, SKa, PKs
	B	B, S, SKb, PKs
	S	A, B, S, SKs, PKa, PKb
2	A	A, B, S, SKa, **PKb**, PKs
	B	B, S, SKb, PKs
	S	A, B, S, SKs, PKa, PKb
3	A	A, B, S, SKa, PKb, PKs, **Na**
	B	**A**, B, S, SKb, PKs, **Na**
	S	A, B, S, SKs, PKa, PKb
4	A	A, B, S, SKa, PKb, PKs, Na
	B	A, B, S, SKb, PKs, Na
	S	A, B, S, SKs, PKa, PKb
5	A	A, B, S, SKa, PKb, PKs, Na
	B	A, B, S, SKb, **PKa**, PKs, Na
	S	A, B, S, SKs, PKa, PKb
6	A	A, B, S, SKa, PKb, PKs, Na, **Nb**
	B	A, B, S, SKb, PKa, PKs, Na, **Nb**
	S	A, B, S, SKs, PKa, PKb
7	A	A, B, S, SKa, PKb, PKs, Na, Nb
	B	A, B, S, SKb, PKa, PKs, Na, Nb
	S	A, B, S, SKs, PKa, PKb

4.3.1.7 Množina aktuálních předpokladů v průběhu protokolu

1	A	→B: B, S, PKs, SKb →S: A, B, S, PKa, SKs
	B	→S: B, S, PKb, SKs
	S	→A: A, B, S, PKs, SKa →B: B, S, PKs, SKb
2	A	→B: B, S, PKs, SKb →S: A, B, S, PKa, **PKb**, SKs
	B	→S: B, S, PKb, SKs
	S	→A: A, B, S, PKs, **PKb**, SKa →B: B, S, PKs, SKb
3	A	→B: **A**, B, S, PKs, SKb, **Na** →S: A, B, S, PKa, PKb, SKs
	B	→A: **A**, **B**, **PKb**, **SKa** →S: B, S, PKb, SKs
	S	→A: A, B, S, PKs, PKb, SKa →B: B, S, PKs, SKb

4	A	→B: A, B, S, PKs, SKb, Na →S: A, B, S, PKa, PKb, SKs
	B	→A: A, B, PKb, SKa →S: **A**, B, S, PKb, SKs
	S	→A: A, B, S, PKs, PKb, SKa →B: **A**, B, S, PKs, SKb
5	A	→B: A, B, S, PKs, SKb, Na →S: A, B, S, PKa, PKb, SKs
	B	→A: A, B, PKb, SKa →S: A, B, S, PKb, **PKa**, SKs
	S	→A: A, B, S, PKs, PKb, SKa →B: A, B, S, PKs, **PKa**, SKb
6	A	→B: A, B, S, PKs, **PKa**, SKb, Na, **Nb** →S: A, B, S, PKa, PKb, SKs
	B	→A: A, B, PKb, SKa, Na, **Nb** →S: A, B, S, PKb, PKa, SKs
	S	→A: A, B, S, PKs, PKb, SKa →B: A, B, S, PKs, PKa, SKb
7	A	→B: A, B, S, PKs, PKa, SKb, Na, Nb →S: A, B, S, PKa, PKb, SKs
	B	→A: A, B, PKb, SKa, Na, Nb →S: A, B, S, PKb, PKa, SKs
	S	→A: A, B, S, PKs, PKb, SKa →B: A, B, S, PKs, PKa, SKb

4.3.1.8 Nedostatky

Tento protokol může útočník přelstít ve dvou relacích tím, že se postupně vydává za klienta A i B. Tím, že získá v první relaci od klienta A nonce Nb, může pak přesvědčit klienta B v druhé relaci, že je důvěryhodný subjekt. Pro úspěšný útok je třeba, aby všichni klienti znali všechny veřejné klíče ostatních klientů [21].

```
 i.3  A  -> I :       {Na, A}KPi
ii.3  Ia -> B :       {Na, A}PKb
ii.6  B  -> Ia:       {Na, Nb}PKa
 i.6  I  -> A :       {Na, Nb}PKa
 i.7  A  -> I :       {Nb}KPi
ii.7  Ia -> B :       {Nb}PKb
```

4.3.1.9 Autor

Roger Needham a Michael Schroeder 1978

4.3.1.10 Publikace

Protokol byl uveden v knize: [6].

Útok je publikován: [21]

4.3.2 Lowe's fixed version of Needham-Schroeder Public Key

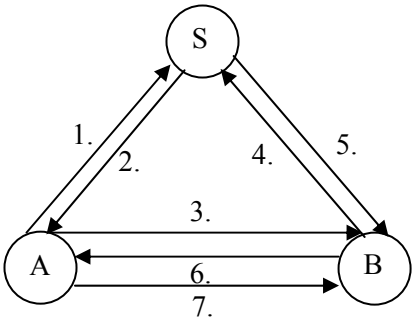
4.3.2.1 Vlastnosti

Protokol Lowe's fixed version of Needham-Schroeder Public Key [21] je opravenou verzí protokolu Needham-Schroeder Public Key, tak aby obstál proti útoku, který byl u něj objeven. Proto je do zprávy v bodu přidána identifikace klienta B, aby bylo jasné, s kterým subjektem je vedena komunikace.

4.3.2.2 Specifikace

```
A, B, S :                    Subjekty
Na, Nb :                     Nonce
PKa,PKb,PKs,SKa,SKb,SKs :    Veřejné a soukromé klíče

1.  A->S:  A,B
2.  S->A:  {PKb, B}SKs
3.  A->B:  {Na, A}PKb
4.  B->S:  B,A
5.  S->B:  {PKa, A}SKs
6.  B->A:  {Na, Nb, B}PKa
7.  A->B:  {Nb}PKb
```



Obr. 23: Schéma komunikace - Lowe's fixed version of Needham-Schroeder Public Key

A – klient
B – klient (klient A chce komunikovat s klientem B)
S – server (server umožní komunikaci. Předá klientovi a klíč pro komunikaci)
PKa, SKa – veřejný a soukromí klíč klienta A
PKb, SKb – veřejný a soukromí klíč klienta B
PKs, SKs – veřejný a soukromí klíč serveru S

4.3.2.3 Množina počátečních znalostí subjektů

A	A, B, S, SKa, PKs
B	B, S, SKb, PKs
S	A, B, S, SKs, PKa, PKb

4.3.2.4 Množina počátečních předpokladů subjektů

A	→B: B, S, PKs, SKb →S: A, B, S, PKa, SKs
B	→S: B, S, PKb, SKs
S	→A: A, S, PKs, SKa →B: B, S, PKs, SKb

4.3.2.5 Množina cílových znalostí

A	PKb, Na
B	PKa, Nb

4.3.2.6 Množina aktuálních znalostí

1	A	A, B, S, SKa, PKs
	B	B, S, SKb, PKs
	S	A, B, S, SKs, PKa, PKb
2	A	A, B, S, SKa, **PKb**, PKs
	B	B, S, SKb, PKs
	S	A, B, S, SKs, PKa, PKb
3	A	A, B, S, SKa, PKb, PKs, **Na**
	B	**A**, B, S, SKb, PKs, **Na**
	S	A, B, S, SKs, PKa, PKb
4	A	A, B, S, SKa, PKb, PKs, Na
	B	A, B, S, SKb, PKs, Na
	S	A, B, S, SKs, PKa, PKb
5	A	A, B, S, SKa, PKb, PKs, Na
	B	A, B, S, SKb, **PKa**, PKs, Na
	S	A, B, S, SKs, PKa, PKb
6	A	A, B, S, SKa, PKb, PKs, Na, **Nb**
	B	A, B, S, SKb, PKa, PKs, Na, **Nb**
	S	A, B, S, SKs, PKa, PKb
7	A	A, B, S, SKa, PKb, PKs, Na, Nb
	B	A, B, S, SKb, PKa, PKs, Na, Nb
	S	A, B, S, SKs, PKa, PKb

4.3.2.7 Množina aktuálních předpokladů v průběhu protokolu

1	A	→B: B, S, PKs, SKb →S: A, B, S, PKa, SKs
	B	→S: B, S, PKb, SKs
	S	→A: A, B, S, PKs, SKa →B: B, S, PKs, SKb
2	A	→B: B, S, PKs, SKb →S: A, B, S, PKa, **PKb**, SKs
	B	→S: B, S, PKb, SKs
	S	→A: A, B, S, PKs, **PKb**, SKa →B: B, S, PKs, SKb
3	A	→B: **A**, B, S, PKs, SKb, **Na** →S: A, B, S, PKa, PKb, SKs
	B	→A: **A**, **B**, **PKb**, **SKa** →S: B, S, PKb, SKs
	S	→A: A, B, S, PKs, PKb, SKa →B: B, S, PKs, SKb
4	A	→B: A, B, S, PKs, SKb, Na →S: A, B, S, PKa, PKb, SKs
	B	→A: A, B, PKb, SKa →S: **A**, B, S, PKb, SKs
	S	→A: A, B, S, PKs, PKb, SKa →B: **A**, B, S, PKs, SKb
5	A	→B: A, B, S, PKs, SKb, Na →S: A, B, S, PKa, PKb, SKs
	B	→A: A, B, PKb, SKa →S: A, B, S, PKb, **PKa**, SKs
	S	→A: A, B, S, PKs, PKb, SKa →B: A, B, S, PKs, **PKa**, SKb

6	A	→B: A, B, S, PKs, **PKa**, SKb, Na, **Nb** →S: A, B, S, PKa, PKb, SKs
	B	→A: A, B, PKb, SKa, Na, **Nb** →S: A, B, S, PKb, PKa, SKs
	S	→A: A, B, S, PKs, PKb, SKa →B: A, B, S, PKs, PKa, SKb
7	A	→B: A, B, S, PKs, PKa, SKb, Na, Nb →S: A, B, S, PKa, PKb, SKs
	B	→A: A, B, PKb, SKa, Na, Nb →S: A, B, S, PKb, PKa, SKs
	S	→A: A, B, S, PKs, PKb, SKa →B: A, B, S, PKs, PKa, SKb

4.3.2.8 Autor

Gavin Lowe 1995

4.3.2.9 Publikace

Protokol byl uveden v knize: [21].

4.3.3 TNM

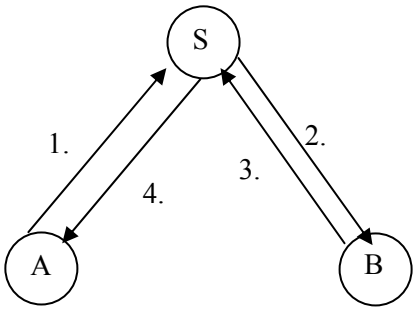
4.3.3.1 Vlastnosti

Protokol TNM [22] zaručuje distribuci symetrického klíče, který bude sloužit ke komunikaci mezi dvěma subjekty. K distribuci se používá třetí strany, a to serveru, který komunikuje s klienty pomocí zpráv, zašifrovanými asymetrickou šifrou (resp. pouze tyto zprávy přijímá). Symetrické klíče se generuji vždy pro jednotlivé relace.

4.3.3.2 Specifikace

```
A, B, S :  Subjekty
Ka, Kb :   Klíče
PKs, SKs : Veřejný a soukromý klíč

1. A->S: B, {Ka}PKs
2. S->B: A
3. B->S: A, {Kb}PKs
4. S->A: B, {Kb}Ka
```



Obr. 24: Schéma komunikace - TNM

A – klient
B – klient (klient A chce komunikovat s klientem B)
S – server, třetí strana
Ka, Kb – vygenerované klíče klientů A a B

4.3.3.3 Množina počátečních znalostí subjektů

A	A, B, S, PKs
B	B, S, PKs
S	A, B, SKs

4.3.3.4 Množina počátečních předpokladů subjektů

A	→B: B, S →S: A, B, S, SKs
B	→S: B, S, SKs
S	→A: A, S, PKs →B: B, S, PKs

4.3.3.5 Množina cílových znalostí subjektů

A	Kb
B	Kb

4.3.3.6 Množina aktuálních znalostí v průběhu protokolu

1	A	A, B, S, PKs, **Ka**
	B	B, S, PKs
	S	A, B, SKs, **Ka**
2	A	A, B, S, PKs, Ka
	B	**A**, B, S, PKs
	S	A, B, SKs, Ka
3	A	A, B, S, PKs, Ka
	B	A, B, S, PKs, **Kb**
	S	A, B, SKs, Ka, **Kb**
4	A	A, B, S, PKs, Ka, **Kb**
	B	A, B, S, PKs, Kb
	S	A, B, SKs, Ka, Kb

4.3.3.7 Množina aktuálních předpokladů v průběhu protokolu

1	A	→B: B, S →S: A, B, S, SKs, **Ka**
	B	→S: B, S, SKs
	S	→A: A, S, PKs, **Ka** →B: B, S, PKs
2	A	→B: B, S →S: A, B, S, SKs, Ka
	B	→A: **A, B, S** →S: A, B, S, SKs
	S	→A: A, S, PKs, Ka →B: **A**, B, S, PKs
3	A	→B: B, S →S: A, B, S, SKs, Ka
	B	→A: A, B, S →S: A, B, S, SKs, **Kb**
	S	→A: A, S, PKs, Ka →B: A, B, S, PKs, **Kb**

4	A	→B: B, S →S: A, B, S, SKs, Ka, **Kb**
	B	→A: A, B, S →S: A, B, S, SKs, Kb
	S	→A: A, S, PKs, Ka, **Kb** →B: A, B, S, PKs, Kb

4.3.3.8 Nedostatky

Velice jednoduché obejití tohoto protokolu, je pomocí podvržení identity, kdy se útočník hned od začátku vydává za klienta A (popř. za klienta B). Jedinou podmínkou je, aby útočník znal veřejný klíč serveru [23].

```
i.1  Ia -> S  :  B, {Ki}PKs
i.2  S  -> B  :  A
i.3  B  -> S  :  A, {Kb}PKs
i.4  S  -> Ia :  B, {Kb}Ki
```

4.3.3.9 Autor

M. Tatebayashi, N. Matsuzaki a D.B. Newman 1989

4.3.3.10 Publikace

Protokol je popsán v publikaci: [22].

Útoky jsou popsány v knize: [23].

5 Přehled bezpečnostních protokolů

Název	Počet subjektů	Distribuce klíče	Autentizace	Třetí strana	Šifrování	Časové razítko
Andrew Secure RPC	2	Ano	Ano	-	Symetrické	Ne
Diffie Helman	2	Ano	Ne	-	Symetrické	Ne
Schnorr's Protokol	2	Ne	Ano	-	Asymetrické	Ne
Needham Schroeder Symmetric Key	3	Ano	Ano	KDC	Symetrické	Ne
Amended Needham Schroeder Symmetric Key	3	Ano	Ano	KDC	Symetrické	Ne
Denning-Sacco shared key	3	Ano	Ne	KDC	Symetrické	Ano
Lowe modified Denning-Sacco shared key	3	Ano	Ano	KDC	Symetrické	Ano
Kao Chow Authentication v.1	3	Ano	Ano	KDC	Symetrické	Ne
Kao Chow Authentication v.2	3	Ano	Ano	KDC	Symetrické	Ne
Wide Mouthed Frog	3	Ano	Ne	KTC	Symetrické	Ano
Yahalom	3	Ano	Ano	KDC	Symetrické	Ne
BAN simplified version of Yahalom	3	Ano	Ano	KDC	Symetrické	Ne
Lowe modified version of Yahalom	3	Ano	Ano	KDC	Symetrické	Ne
Otway Rees	3	Ano	Ano	KDC	Symetrické	Ne
KSL	3	Ano	Ano	KDC	Symetrické	Ano
Lowe modified KSL	3	Ano	Ano	KDC	Symetrické	Ano
Woo and Lam Pi	3	Ne	Ano	-	Symetrické	Ne
Needham-Schroeder Public Key	3	Ano	Ano	KTC	Asymetrické	Ne
Lowe's fixed version of Needham-Schroeder Public Key	3	Ano	Ano	KTC	Asymetrické	Ne
TNM	3	Ano	Ne	KTC	Asymetrické	Ne

6 Závěr

Cílem této bakalářské práce bylo shromáždění informací o nejznámějších bezpečnostních protokolech. Na základě získaných údajů pak vytvořit přehled těchto protokolů v písemné a elektronické podobě.

Během mého studia bezpečnostních protokolů jsem po konzultaci s vedoucím práce změnil bod 4 zadání tak, aby externí aplikace přistupovaly přímo do databáze protokolů. Toto řešení je efektivnější a jednodušší. Hlavním přínosem je specifikace protokolů a detailní rozbor jejich chování. Specifikace byla provedena na základě množin znalosti a předpokladů, které lze velmi jednoduše transformovat do logik množin a znalostí (logics of knowledge a beliefs) - například BAN logika. Zpracování protokolů v těchto logikách by však již bylo nad rámec mé práce. Zkoumání chování a specifikace těchto protokolů vedlo k podrobnému rozboru a pochopení protokolů, a to nejen z mé strany, ale nabízí ucelený přehled i čtenáři této práce. Uvedení specifikací a popisu chování ve všech stavech lze považovat také za praktický přínos a tímto vedlo k většímu rozsahu písemné části.

Pro snazší dostupnost výsledků mé práce jsou všechny zpracované protokoly zveřejněny na internetu. Vytvořil jsem databázi bezpečnostních protokolů. Neuvádím v ní pouze specifikace, ale celou analýzu. Podle zadání je možné z této databáze použít specifikace jednotlivých protokolů, které jsou potřebné pro verifikaci bezpečnostních protokolů. Kromě této vlastnosti je možné do databáze dále přidávat nové protokoly a opravovat údaje již vložené, podle komentářů, které je možno přidávat k jednotlivým protokolům.

V budoucnu je možné tuto databázi dál vylepšovat a vytvářet další analýzy bezpečnostních protokolů. Přestože mým cílem bylo vytvořit český přehled bezpečnostních protokolů, databáze je připravena na vícejazyčnost. Tento informační systém by se také mohl rozšířit o základní zdrojové kódy pro jednotlivé verifikační jazyky a tím ulehčit externím aplikacím spolupráci se systémem.

Literatura

[1] Satyanarayanan, M. Integrating security in a large distributed system. ACM Transactions on Computer Systems, 1989.

[2] Burrows, M., Abadi, M., Needham, R. A logic of authentication. Technical Report 39, Digital Systems Research Center, february 1989.

[3] Diffie, W., Helman, M. New directions in cryptography. IEEE Transactions on Information Society, 1976.

[4] Blanchet, B. An efficient cryptographic protocol verifier based on prolog rules. In IEEE, editor, 14th IEEE Computer Security Foundations Workshop (CSFW-14), 2001.

[5] Cramer, R., Damgård, I., Schoenmakers, B. Proofs of partial knowledge and simplified design of witness hiding protocols. In Proc. 14th Annual International Cryptology Conference (CRYPTO'94), Santa Barbara (California, USA), 1994.

[6] Needham, R., Schroeder, M. Using encryption for authentification in large networks of computers. Communications of the ACM, December 1978.

[7] Denning, D., Sacco, G. Timestamps in key distributed protocols. Communication of the ACM, 1981.

[8] Needham, R., Schroeder, M. Authentication revisited. Operating Systems Review, 1987.

[9] Lowe, G. A family of attacks upon authentication protocols. Technical Report 1997/5, Department of Mathematics and Computer Science, University of Leicester, 1997.

[10] Kao, I L., Chow, R. An efficient and secure authentication protocol using uncertified keys. Operating Systems Review, 1995.

[11] Burrows, M., Abadi, M., Needham, R. A logic of authentication. Technical Report 39, Digital Systems Research Center, february 1989.

[12] Paulson, L. C. Relations between secrets: Two formal analyses of the yahalom protocol. J. Computer Security, 2001.

[13] Syverson, P. A taxonomy of replay attacks. In Proceedings of the 7th IEEE Computer Security Foundations Workshop. IEEE Computer Society Press, 1994.

[14] Lowe, G. Towards a completeness result for model checking of security protocols. Technical Report 1998/6, Dept. of Mathematics and Computer Science, University of Leicester, 1998.

[15] Otway, D., Rees, O. Efficient and timely mutual authentication. Operating Systems Review, 1987.

[16] Clark, J., Jacob, J. A survey of authentication protocol literature : Version 1.0., 1997.

[17] Kehne, A., Schönwälder, J., Langendörfer, H. Multiple authentications with a nonce-based protocol using generalized timestamps. In Proc. ICCC ’92, Genua, 1992.

[18] Lowe, G. Some new attacks upon security protocols. In IEEE Computer Society Press, editor, In Proceedings of the Computer Security Foundations Workshop VIII, 1996.

[19] Woo, T. Y. C., Lam, S. S. A lesson on authentication protocol design. Operating Systems Review, 1994.

[20] Hwang, T., et al. Two attacks on neumannstubblebine authentication protocols. Information Processing Letters, 1995.

[21] Lowe, G. An attack on the Needham-Schroeder public key authentication protocol. Information Processing Letters, 1995.

[22] Tatebayashi, M., Matsuzaki, N., Newman, D. B. Key distribution protocol for digital mobile communication systems. In Advance in Cryptology --- CRYPTO '89. 1989.

[23] Lowe, G., Roscoe, A. W. Using CSP to detect errors in the TMN protocol. Software Engineering, 1997.

[24] Ptáček, T. Specifikační jazyky a nástroje pro analýzu a verifikaci bezpečnostních protokolů. Brno, 2007. 94 s. Vedoucí diplomové práce Pavel Očenášek.

[25] Očenášek, P. Využití stromových automatů při verifikaci bezpečnostních komunikačních protokolů. Brno, 2004. 17 s. FIT VUT v Brně. Oborová práce.

[26] *Security Protocols Open Repository* [online]. [2002-2003] [cit. 2008-05-01]. Dostupný z WWW: <http://www.lsv.ens-cachan.fr/spore/index.html>.

[27] *ProtocolLounge* [online]. [2007] [cit. 2008-05-01]. Dostupný z WWW: <https://wiki.isi.qut.edu.au/ProtocolLounge/doku.php>.

[28] Mark, A.. *The CTDP Security Reference Manual* [online]. 2007 [cit. 2008-05-01]. Dostupný z WWW: <http://www.comptechdoc.org/independent/security/guide/secprotocols.html>.

[29] Avispa. *The AVISPA Library of protocols* [online]. [2007] [cit. 2008-05-01]. Dostupný z WWW: <http://www.avispa-project.org/library/index.html>.

Seznam obrázků

Obr. 1: Útok - Přerušení komunikace 4
Obr. 2: Útok - Podvržení identity 4
Obr. 3: Útok - Odposlouchávání 4
Obr. 4: Útok - Modifikace zprávy 4
Obr. 5: Schéma komunikace - Andrew Secure RPC 7
Obr. 6: Schéma komunikace - Diffie helman 9
Obr. 7: Schéma komunikace - Schorr's protokol 11
Obr. 8: Schéma komunikace - Needham Schroeder Symetric Key 13
Obr. 9: Schéma komunikace - Amended Needham Schroeder Symetric Key 16
Obr. 10: Schéma komunikace - Denning-Sacco shared key 18
Obr. 11: Schéma komunikace - Lowe modified Denning-Sacco shared key 20
Obr. 12: Schéma komunikace - Kao Chow Authentication v.1 22
Obr. 13: Schéma komunikace - Kao Chow Authentication v.2 24
Obr. 14: Schéma komunikace - Wide Mouthed Frog 26
Obr. 15: Schéma komunikace - Yahalom 28
Obr. 16: Schéma komunikace - BAN simplified version of Yahalom 30
Obr. 17: Schéma komunikace - Lowe modified version of Yahalom 33
Obr. 18: Schéma komunikace - Otway Rees 35
Obr. 19: Schéma komunikace - KSL 37
Obr. 20: Schéma komunikace - Lowe modified KSL 41
Obr. 21: Schéma komunikace - Woo And Lam Pi 44
Obr. 22: Schéma komunikace - Needham-Schroeder Public Key 46
Obr. 23: Schéma komunikace - Lowe's fixed version of Needham-Schroeder Public Key 49
Obr. 24: Schéma komunikace - TNM 51

Seznam příloh

Příloha 1. – Manuál k databázi bezpečnostních protokolů

Příloha 2. – CD se zdrojovými texty a písemnou částí v PDF

Příloha 1. – Manuál k databázi bezpečnostních protokolů

Motivace

Cílem této databáze je vytvořit přehledný soubor bezpečnostních protokolů, které jsou zpracovány jednotným způsobem. Takhle vytvořenou databázi pak zpřístupnit[3] pro veřejnost, která se o tuto problematiku zajímá. Proto je kladen požadavek, aby specifikace všech protokolů byla zaznamenána v databázi v původní formě a v předem dohodnuté a jednoznačné syntaxi, aby bylo možné tuto informaci dále použít pro verifikaci bezpečnostních protokolů externími aplikacemi.

Implementace

Databáze je navrhnuta a implementována v prostředí MySQL a běží na eva.fit.vutbr.cz . Samotná databáze se skládá pouze z jedné relace a tou je samotný protokol. Atributy s datovým typem text jsou vyjma specifikace přímo formátovány jako HTML tabulky, pro snadné zobrazení. Požadavek na zachování čistého textu se vztahuje pouze na atribut specifikace, s kterým se počítá dále pracovat, na rozdíl od zbytku, který slouží pouze jako prezentace daného bezpečnostního protokolu. Jednotlivé atributy a jejich vlastnosti lze vidět v následující tabulce:

Sloupec	**Typ**	**Nulový**	**Výchozí**
nazev	varchar(60)	Ne	
vlastnosti	text	Ano	NULL
specifikace	text	Ne	
specifikace_poz	text	Ano	NULL
poc_znalosti	text	Ne	
poc_predpoklady	text	Ne	
cil_znalosti	text	Ne	
akt_znalosti	text	Ne	
akt_predpoklady	text	Ne	
nedostatky	text	Ano	NULL
autor	varchar(30)	Ano	NULL
rok	year(4)	Ano	NULL
publikace	text	Ano	NULL
pocet	int(11)	Ne	0
distribuce	varchar(3)	Ne	
autentizace	varchar(3)	Ne	
server	varchar(3)	Ne	
sifrovani	varchar(12)	Ne	
razitko	varchar(3)	Ne	
komentar	text	Ano	NULL

[3] Databáze je umístěna na internetové stránce www.stud.fit.vutbr.cz/~xpapez01/ibp

Soubory

Informační systém je naprogramován v PHP a obsahuje tyto soubory:

index.php	Úvodní stránka, na které jsou základní informace o zpracování protokolů. Také je zde jediná možnost se přihlásit a odhlásit pro administrátora[4].
Prehled.php	Tento skript vytváří přehled bezpečnostních protokolů. Uživatel si může všechny protokoly s jejich specifikacemi exportovat do XML nebo si prohlédnou detaily protokolu tím, že klikne na jeho jméno v tabulce. Administrátor je schopen vybraný protokol smazat.
detail.php	Skript, který vypíše vše o vybraném protokolu. Na závěr je možnost vytvořit komentář k tomuto protokolu, pokud se objeví chyba v analýze protokolu. Komentář může vytvořit kdokoli, ale smazat jedině přihlášený administrátor.
VlozProtokol.php	Stránka pro uložení nového bezpečnostního protokolu. Nejprve si administrátor vybere kolik subjektů a v kolika pravidlech protokol pracuje a dále podle těchto údajů se vygeneruje formulář pro všechny údaje, které jsou uloženy v databázi.
Html.php	Vytvoří hlavičku a tělo HTML stránky.
Texty.php	Obsahuje všechny použité textové řetězce, které jsou použity na stránkách. Tvoří základ a polotovar pro vytvoření dalších jazykových mutací tohoto systému.
Styly.css	Použité styly.
Protokoly.xml	Implicitní soubor pro export protokolů.

Parametry

Počet souborů: 8

Velikost: 68 kB

Počet řádků: 1830

Testováno – OS: Microsoft Windows Vista a XP

– prohlížeč: Internet Explorer 7 a Mozilla Firefox

[4] Pro přihlášení jako administrátor je nutno použít login: admin a heslo: lolek