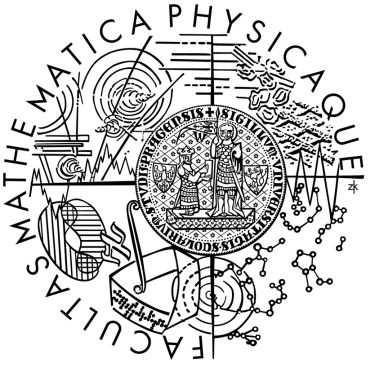
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Marie Švarcová

Konkrétní bezpečnost protokolu IPSec

Katedra algebry

Vedoucí diplomové práce: RNDr. Bohuslav Rudolf

Studijní program: Matematika
Studijní obor: Matematické metody informační bezpečnosti

Praha 2015

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne ..... Marie Švarcová

Název práce: Konkrétní bezpečnost protokolu IPSec

Autor: Marie Švarcová

Katedra: Katedra algebry

Vedoucí diplomové práce: RNDr. Bohuslav Rudolf, Katedra algebry

Abstrakt: Hlavním cílem této práce je formulace a důkaz bezpečnostních vlastností protokolu IKE, jehož prostřednictvím dochází k dohodě na klíčích používaných protokolem IPSec k zabezpečení síťové komunikace. Věnujeme se také popisu rozdílů konkrétní a asymptotické bezpečnosti a přinášíme definice pojmu bezpečnosti výměny klíče a základních kryptografických primitivů používaných protokoly výměny klíče právě v kontextu konkrétní bezpečnosti. Dále přinášíme obecný popis protokolu IPSec a jeho hlavních funkcionalit následovaný podrobným popisem obou verzí protokolu IKE. Součástí práce je úvod do problematiky protokolů výměny klíče a popis a analýza protokolu Sigma0, který tvoří jádro verze protokolu IKE používající k ochraně autentičnosti a integrity digitální podpis.

Klíčová slova: IPSec, IKE, konkrétní bezpečnost, výměna klíčů s ochranou autentičnosti a integrity

Title: Concrete Security of the IPSec Protocol

Author: Marie Švarcová

Department: Department of Algebra

Supervisor: RNDr.Bohuslav Rudolf, Department of Algebra

Abstract: The main goal of this thesis is to articulate and to prove security properties of the key exchange protocol IKE, through which the IPSec protocol establishes agreement on keys used for securing internet traffic. It also covers the description of differences between asymptotic and concrete security treatments and the notions of key exchange security and the security of underlying primitives used by key exchange protocols, in the context of concrete security. A general description of IPSec and its main functionalities follows, accompanied by detailed descriptions of both versions of IKE (IKEv1, IKEv2). A general introduction to key exchange is also included and a representative of signature-based version of IKE is introduced and its security is analysed.

Keywords: IPSec, IKE, concrete security, authenticated key exchange

Obsah

Kapitola 1

Úvod

Cílem této práce je seznámení se s protokolem IPSec, který je v praxi hojně používaným protokolem pro zajištění bezpečnosti síťových komunikací, a to zejména s protokolem IKE, na němž bezpečnost této komunikace stojí. Jedná se o protokol výměny klíče, který zajišťuje výměně ochranu autentičnosti a integrity.

Dalším cílem pak je představit důkaz konkrétní bezpečnosti tohoto protokolu. Vycházíme přitom z již existujícího důkazu, který je však vyjádřen pouze z pohledu asymptotické bezpečnosti. Použity ale byly pouze argumenty představeného důkazu. Explicitní vyjádření bezpečnosti je již samostatným dílem autora.

1.1 Přehled představených témat a použité zdroje

V úvodní části práce nejprve definuje rozdíl mezi asymptotickou a konkrétní bezpečností. Použité definice jsme s mírnou úpravou převzali z [5].

Následuje kapitola věnovaná protokolu IPSec. Představíme si obecné fungování protokolu a používané prvky. Součástí bude také stručný popis shémat zajišťujících ochranu přenášených paketů (AH a ESP). Významná část kapitoly je ale věnována podrobnému popisu protokolu výměny klíče IKE a jeho variant IKEv1 a IKEv2. Čerpáme především z [11], [7], [3], [10], [9], [4], [6].

Ve třetí kapitole popisujeme modelové prostředí protokolu výměny klíče a definujeme základní pojmy použité v protokolu. Definujeme schopnosti a akce útočníka na protokol a také jeho cíle. Zde také definujeme pojem bezpečnosti protokolu výměny klíče, kterou nazýváme SK-bezpečnost. Tato definice je zde popsána z pohledu asymptotické bezpečnosti. Zde vycházíme především z [1], [2], [8].

Čtvrtá kapitola je věnována popisu protokolu Σ_0. Popisujeme zde akce jednotlivých účastníků a přinášíme hlavní myšlenku důkazu jeho bezpečnosti. V poslední části se věnujeme představení simulátorů, které tvoří zásadní stavební kameny celého důkazu. Tato část je převzata z [2].

V páté kapitole již přecházíme ke konkrétní bezpečnosti. Znovu uvádíme základní pojmy vztahující se k protokolům výměny klíče, tentokrát však v kontextu konkrétní bezpečnosti. Přidáváme definice základních kryptografických primitivů a ostatních důležitých prvků použitých v důkaze.

V šesté kapitole již přinášíme podrobný důkaz konkrétní bezpečnosti protokolu Σ_0 a tyto výsledky formulujeme do věty 1. Kapitola je rozdělena do dvou části a každá z nich se věnuje důkazu konkrétní vlastnosti definice.

V závěru pak přinášíme shrnutí získaných výsledků a popisujeme, jak je aplikovat na schéma IKE.

1.2 Asymptotická vs. konkrétní bezpečnost

Ještě než přistoupíme k popisu protokolu IPSec a jeho konkrétních verzí protokolů výměny klíče, vysvětlíme si, jaký je rozdíl mezi asymptotickou a konkrétní bezpečností.

Jedná se vlastně o dva přístupy k zacházení s bezpečnostními pojmy. Oba tyto přístupy spadají do kategorie výpočetní (nebo také dokazatelné) bezpečnosti. To je kategorie, která je orientovaná na praktické („de-facto“) výsledky. Nepředpokládá absolutní bezpečnost konkrétního algoritmu nebo schématu, ale zajímá se o omezení výpočetní síly útočníka. Pro každého útočníka předpokládá určitou (malou) pravděpodobnost úspěchu útoku. Na druhou stranu ale uvažuje pouze efektivní útočníky, kteří pracují v nějakém dosažitelném čase.

Asymptotická bezpečnost Asymptotický přístup používá k ohodnocení míry bezpečnosti schématu celé číslo n, které se nazává *bezpečnostní parametr*. Pomocí něj jsou vyjádřeny jak vlastnosti schématu, tak vlastnosti jeho uživatelů (včetně útočníků). Výpočetní zdroje i pravděpodobnost úspěchu útočníka jsou funkce bezpečnostního parametru, ne konkrétní hodnoty.

Efektivním útočníkem je algoritmus, který běží v *polynomiálním* čase. To znamená, že čas běhu algoritmu, který reprezentuje chování útočníka, je omezen polynomem $p(n)$ pro nějaký bezpečnostní parametr n. „Malá“ pravděpodobnost úspěchu je taková, která je menší než libovolný inverzní polynom $\frac{1}{p(n)}$. Používá se pro ni pojem *zanedbatelá*. Řečí asymptotické bezpečnosti je určité schéma bezpečné, pokud pro libovolného polynomiálního

útočníka platí, že uspěje v prolomení schématu s nejvýše zanedbatelnou pravděpodobností.

Na bezpečnostní parametr můžeme nahlížet jako na prostředek umožňující nastavit požadovanou míru bezpečnosti. Ovšem je třeba si uvědomit, že s rostoucí velikostí bezpečnostního parametru neroste pouze míra bezpečnosti, ale také výpočetní zdroje potřebné pro provoz protokolu. (Operace nejsou náročnější pouze pro útočníka, ale zároveň i pro uživatele protokolu).

Konkrétní bezpečnost Na rozdíl od asymtotické bezpečnosti dovoluje přistup konkrétní bezpešnosti explicitním omezením maximální pravděpodobnosti úspěchu libovolného útočníka kvantifikovat míru bezpečnosti. Zatímco z asymptotického pohledu schéma buď je, nebo není bezpečné, z pohledu konkrétní bezpečnosti můžeme říci, že jedno schéma je bezpečnější než druhé, a máme k dispozici konkrétní bezpečnostní redukce (lze tedy ohodnotit jejich kvalitu). Pokud je redukce silná, bude stačit kratší klíč a protokol se tak stane efektivnější. Konkrétní garance bezpečnosti jsou přesně tím, na čem v praxi záleží. Navíc se snaží zachovat co nejvíce síly ve schématu použitých primitivů[1].

Z pohledu konkrétní bezpečnosti je vždy třeba nejprve definovat, co to znamená prolomit bezpečnost. Obecný postup důkazu vypadá tak, že pro hodnoty vyjadřující výpočetní schopnosti a pravděpodobnost úspěchu útočníka dokážeme nějaké omezující vztahy, které vyjádříme pomocí použitých bezpečnostních předpokladů.

Poznámka Pro lepší pochopení následující kapitoly je třeba alespoň základní znalost počítačových sítí a architektury TCP/IP. Podrobný přehled lze nalézt například v [7].

[1] Jedná se o základní stavební kameny kryptografických schémat. Jsou to dále nedělitelné prvky, do kterých patří symetrické a asymetrické šifry, hashovací funkce, pseudonáhodné funkce a generátory apod.

Kapitola 2

Protokol IPSec

Protokol IPSec (Internet Protocol Security) vznikl jako řešení zásadního nedostatku protokolu IP (Internet Protocol), kterým je absence mechanismů pro zajištění bezpečnosti síťových komunikací. Tato bezpečnost zahrnuje tři oblasti. Jsou to zajištění autentičnosti (příjemce zprávy si může ověřit, že předpokládaný původce zprávy je skutečně odesilatelem) a integrity (příjemce zprávy si může ověřit, že během přenosu nedošlo k pozměnění zprávy) přenášených zpráv, utajení jejich obsahu a také správa klíčů. Zjednodušeně řečeno IPSec funguje tak, že dva subjekty, které chtějí bezpečně komunikovat v rámci nějaké „otevřené“ sítě, se nejprve bezpečným způsobem dohodnou na sdíleném tajném klíči a s jeho využitím pak zabezpečí následnou komunikaci. Samozřejmě spolu s klíčem si obě strany musí dohodnout také rozsah zabezpečení (např. zda požadují utajení, nebo stačí ochrana autentičnosti a integrity zpráv) a konkrétní kryptografické algoritmy, které použijí. Hlavní výhodou IPSec je to, že dokáže zabezpečit veškerou komunikaci na IP vrstvě, tedy funguje pro libovolné aplikace.

IPSec dokumenty IPSec není definován jako jeden internetový standard, ale jeho achritektura, služby a konkrétní protokoly jsou popsány souborem dokumentů RFC. Tento soubor se nachází v dokumentu [3]. Tyto dokumenty můžeme rozdělit do několika skupin. První skupinu tvoří dokumenty týkající se architektury IPSec. Ty obsahují základní koncepty, bezpečnostní požadavky a definice technologií IPSec. Druhou skupinu tvoří protokoly *Authentication Header (AH)* a *Encapsulating Security Payload (ESP)*, které zabezpečují přenášené IP pakety („běžnou“ komunikaci po síti). Třetí skupinu tvoří dokumenty zabývající se mechanismy správy klíče, které jsou popsány protokolem *Internet Key Exchange (IKE)*. Čtvrtou skupinu tvoří dokumenty popisující kryptografické algoritmy a poslední pátou skupinu tvoří ostatní dokumenty vztahující se k IPSec.

Poznámka V následující sekci budou opakovaně zmiňovány protokoly AH, ESP a IKE. Jejich popis bude následovat v sekcích 2.2, 2.3 a 2.4. Pokud bychom je zahrnuli do našeho zjednodušeného příkladu z úvodu, pak IKE je ten protokol, který realizuje dohodu na klíči, rozsahu zabezpečení a konkrétních algoritmech, a protokoly AH a ESP (samostatně nebo v kombinaci) realizují zabezpečení následné komunikace pomocí dohodnutých prvků a v dohodnutém rozsahu. Ještě je důležité zmínit, že IPSec pracuje ve dvou módech, jsou to *Transport Mode* a *Tunnel Mode*. Ty ovlivňují, jakým způsobem protokoly AH a ESP „chrání“ jednotlivé pakety (neboli jakým způsobem vytvářejí z nechráněných IP paketů chráněné IPSec pakety). Obecně platí, že Transport Mode poskytuje ochranu protokolům vyšších vrstev, tedy zapouzdřuje pouze payload IP paketu, zatímco Tunnel Mode poskytuje ochranu celému paketu (to funguje tak, že po přidání AH nebo ESP položek k původnímu paketu je takto vzniklý celek vnímán jako payload nového vnějšího IP paketu, ke kterému je přidána nová IP hlavička).

2.1 Bezpečnostní politiky, bezpečnostní asociace a selektory

Protože zajištění bezpečnosti paketů zahrnuje určitou režii, nebylo by moc efektivní, kdyby se zabezpečení odvozovalo pro každou odchozí a příchozí zprávu zvlášť. Z tohoto důvodu má IPSec mechanismus, kterým určuje, jak se bude nakládat s jednotlivými typy paketů. K tomu slouží následující tři koncepty.

Prvním z nich je *bezpečnostní politika (security policy)*. Jedná se o určité pravidlo, které IPSecu říká, jak má zpracovávat IP pakety (např. zda ho má zabezpečit a jakým způsobem). Jednotlivé bezpečnostní politiky se ukládají do *databáze bezpečnostních politik (Security Policy Database - SPD)*.

Druhým konceptem je *bezpečnostní asociace (Security Association - SA)*. Je to logické spojení mezi odesilatelem a příjemcem, které popisuje konkrétní mechanismy, které jsou použity pro zajištění bezpečnosti komunikace mezi nimi. Každá SA je jednoznačně identifikována trojicí parametrů *SPI* (Security Parameter Index - bitový řetězec lokálního významu jednoznačně identifikující SA), IP adresa příjemce a identifikátor bezpečnostního protokolu (AH nebo ESP). Bezpečnostní asociace jsou uloženy v *databázi bezpečnostních asociací (Security Association Database - SAD)*.

Posledním konceptem je *selektor (selector)*. Pomocí selektorů se odvozuje, jakou politiku (respektive jakou SA) použít pro konkrétní paket. Selektor je množina pravidel definovaných každou SA, podle kterých se volí pakety, které

se jí týkají. Může být např. definován tak, že konkrétní SA bude pro paket použita, pokud pro určitou hodnotu cílové adresy bude hodnota zdrojové adresy v nejakém definovaném rozmezí.

Zpracování paketů protokolem IPSec V praxi to tedy vypadá tak, že každý odchozí IP paket je zpracován IPSecem předtím, než je předán k odeslání nižší vrstvě, a každý příchozí IP paket je zpracován IPSecem předtím, než je jeho obsah předán vyšší vrstvě. Zpracování odchozího paketu probíhá tak, že nejprve se porovnají selektory obsažené v paketu s SPD a najde se příslušný záznam, který dále odkazuje na nula nebo více SA (pokud se žádný záznam nenajde, paket se zahodí). Poté se dle konkrétní politiky paket buď zahodí, předá nižší vrstvě k odeslání, nebo se zabezpečí. Pokud má dojít k zabezpečení, najde se v SAD příslušná SA a paket je podle ní zpracován protokolem AH nebo ESP a předán k odeslání. Pokud se záznam nenajde, zavolá se protokol IKE a je vytvořena nová SA, přidána do SAD. Pro příchozí paket je situace podobná. Nejprve se zjistí, zda se jedná o zabezpečený paket. Pokud ne, najde se příslušný záznam v SPD a paket se buď zahodí, nebo je předán vyšší vrstvě. Pokud je paket zabezpečený, najde se příslušná SA v SAD, paket je podle ní zpracován a předán vyšší vrstvě. Pokud se záznam nenajde, je paket zahozen.

2.2 Authentication Header

Protokol AH zajišťuje ochranu autentičnosti a integrity paketů, ale nezajišťuje utajení přenášených dat. Volitelně také zajišťuje ochranu proti replay útokům[1]. Jak již vyplývá z názvu, ochrana je poskytnuta vytvořením AH hlavičky, která je přidána k původnímu IP paketu. AH hlavička se skládá z několika polí, z nichž nejdůležitější je pole *Authentication Data*, které obsahuje hodnotu ICV (Integrity Check Value). Tato hodnota se počítá pomocí kryptografického hashovacího algoritmu na základě většiny položek původního paketu (nepoužijí se ty, které se během přenosu mění).

Právě ICV poskytuje zmíněnou ochranu autentičnosti a integrity zprávy. K jejímu výpočtu je totiž použit sdílený tajný klíč a to příjemci zajišťuje, že pokud pomocí tohoto klíče na základě přijatých dat vypočítal stejnou hodnotu ICV, která je obsažena v položce Authentication Data, pak odesilatelem zprávy je účastník, se kterým proběhla dohoda na klíči, a zpráva nebyla během přenosu pozměněna.

[1]Jedná se o typ útoku, při kterém se útočník snaží obelstít protistranu tak, že znovu posílá zachycené validní zprávy z nějaké předchozí komunikace tohoto účastníka.

Vytvoření nového AH paketu Nový zabezpečený paket se sestaví podle požadovaného módu. Pro Transport Mode je AH hlavička vložena mezi původní, odpovídajícím způsobem pozměněnou (k původní délce paketu se přičte délka AH hlavičky a typ protokolu je AH) IP hlavičku a původní payload. Pro Tunnel Mode je celý původní IP paket zachován, před něj je umístěna AH hlavička (nyní počítaná ze všech položek původního paketu) a toto dohromady odpovídá payloadu nového paketu. K němu je nově vytvořena a přidána odpovídající IP hlavička.

2.3 Encapsulating Security Payload

Protokol ESP zajišťuje utajení přenášených dat a volitelně může zajistit také ochranu jejich autentičnosti a integrity nebo ochranu proti replay útokům. Utajení je dosaženo zašifrováním části nebo celého (v závislosti na módu) původního IP paketu s použitím sdíleného tajného klíče. Ochrana autentičnosti a integrity je zajišťena podobně jako v případě protokolu AH, tedy připojením hodnoty ICV počítané pomocí kryptografického hashovacího algoritmu.

Vytvoření nového ESP paketu ESP je implementován pomocí tří komponent. Jsou to *ESP Header*, *ESP Trailer* a volitelná *ESP Authentication Data*, obsahující hodnotu ICV. Nový paket je opět sestaven v závislosti na módu. Obecný postup je takový, že v prvním kroku je vytvořena a umístěna ESP hlavička. Ve druhém kroku je vypočten ESP Trailer, je připojen za zabezpečovaná data a tento blok (data + trailer) je zašifrován a připojen za ESP hlavičku. ESP Trailer má především funkci paddingu, tedy jeho připojením za data vznikne blok vhodné délky pro použití šifrovacího algoritmu. Třetí (volitelný) krok představuje výpočet hodnoty ICV přes takto vzniklý celek a její připojení za něj. Pro Transport Mode je nový paket tvořen původní (mírně pozměněnou) IP hlavičkou následovanou zašifrovaným blokem a volitelně hodnotou ICV. Pro Tunnel Mode tvoří zabezpečovaná data celý původní IP paket, proto se pro nový paket vypočte nová IP hlavička, za kterou opět následuje zašifrovaný blok a volitelně hodnota ICV.

2.4 Internet Key Exchange

Protokol IKE představuje tu část protokolu IPSec, která řeší správu klíče (především odvození a distribuci sdílených klíčů) a ostatních parametrů. Typickým požadavkem pro internetovou komunikaci v praxi jsou čtyři klíče,

jedna dvojice pro zajištění utajení příchozích a odchozích zpráv a druhá dvojice pro ochranu jejich autentičnosti a integrity (pro každý směr komunikace jiný klíč). Je to právě IKE, který poskytuje automatické mechanismy pro ustanovení a správu SA v souladu s politikami v SPD a pro vytváření příslušných klíčů. V současné době existuje IKE ve dvou verzích, jsou to původní *IKEv1* a nový revidovaný *IKEv2*. Základní funkcionalita ale zůstává zachována.

Odvození klíče v IKE je vylepšením Diffie-Hellmanova algoritmu (DH). Oproti němu poskytuje navíc ochranu proti replay útokům použitím noncí, proti clogging útokům použitím cookies a proti útokům man-in-the-middle (MITM) zajištěním ochrany autentičnosti a integrity přenášeným zprávám.[2]

Nonce a cookies Nonce je lokálně generované číslo, které se použije pouze jednou. V určitých částech protokolu jsou nonce zašifrované, aby bylo jejich použití bezpečné. Cookie je pseudonáhodné číslo, které mohou oba komunikující poslat v inicializační zprávě protokolu. Příjemce ho musí potvrdit a musí být zopakováno v první zprávě samotné DH výměny. IKE nařizuje, aby cookies splňovaly tři základní podmínky: 1. musí záviset na konkrétních účastnících (aby ho útočník nemohl získat); 2. pro jiného účastníka, než je ten, který cookie vydal, nesmí být možné vytvořit cookie, které vydávající účastník přijme (účastník musí při generování použít nějakou lokální tajnou informaci, kterou nesmí být možno z žádné konkrétní cookie odvodit); 3. generování a ověřování musí být rychlé, aby se předešlo útokům sabotáží výpočetních zdrojů.

Metody zajišění ochrany autentičnosti a integrity V rámci protokolu IKE mohou být použity tři různé metody zajištění ochrany autentičnosti a integrity:

- **digitální podpisy** - ochranu poskytuje podepsání vzájemně dosažitelného hashe důležitých položek (např. ID, nonce)
- **šifrování s veřejným klíčem** - ochranu poskytuje zašifrování důležitých položek soukromým klíčem odesilatele
- **symetrické šifrování** - ochranu poskytuje zašifrování položek klíčem, který je odvozen nějakým jiným kanálem

[2]Clogging útok je takový, při kterém útočník zahltí systém účastníka požadavky na výměnu klíče s falešnými adresami. Účastník pak zbytečně provádí nákladné výpočty DH hodnot. Při MITM útočník pozměňuje zprávy posílané v rámci určité komunikace bez vědomí komunikujících.

Formát IKE paketů IKE definuje formáty paketů pro různé akce týkající se SA (např. ustanovení, smazání, modifikace). Ty poskytují konzistentní rámec nezávislý na konkrétních algoritmech. Každý paket se skládá z IKE hlavičky následované jedním nebo více payloady (konkrétně definovanými pro každou akci).

2.4.1 IKEv1

IKEv1 není popsán v jednom dokumentu, ale tvoří ho společně dokumenty [10] (IPSec DOI), [9] (ISAKMP) a [4] (IKE).[3] Poslední z dokumentů je ale stěžejní.

Protokol probíhá ve dvou fázích. V první fázi (*Phase 1*) se dva účastníci dohodnou, jakým způsobem budou dál bezpečně komunikovat. Výstupem této fáze je bezpečnostní asociace IKE SA, která se použije ve druhé fázi protokolu (*Phase 2*) pro bezpečné ustanovení SA pro protokoly AH nebo ESP. První fáze navíc může probíhat ve dvou módech, v *hlavním módu (Main Mode)* nebo v *agresivním módu (Aggresive Mode)*. Druhá fáze se také někdy nazýva *rychlý mód (Quick Mode)*. Je to proto, že tato fáze může být podle potřeby provedena několikrát s použitím již dohodnuté IKE SA, nebo může být v jednom rychlém módu dohodnuto několik SA (např. když účastníci chtějí v rámci jedné komunikace posílat zprávy s různým typem zabezpečení). Protože může každý účastník v jednu chvíli vést najednou několik různých komunikací, každá zpráva obsahuje na začátku identifikátor spojení, který určuje, ke které z těchto komunikací patří. Tento identifikátor se skládá z dvojice (cookie iniciátora, cookie respondéra) a je jednoznačný.

Phase 1 - Main Mode V hlavním módu se první fáze skládá ze tří výměn (celkem 6 zpráv). To umožňuje skrýt identity komunikujících účastníků. Každá výměna probíhá tak, že iniciátor komunikace pošle zprávu a respondér pošle odpověď. Další výměna nezačne, dokud nebyla dokončena předchozí. V první výměně jsou dohodnuty bezpečnostní parametry. Iniciátor pošle nabídku jím podporovaných kryptografických sad a respondér odpovídá, kterou si zvolil. Ve druhé výměně si účastníci vymění DH hodnoty a pomocná data (např. nonce). V tuto chvíli oba účastníci odvodí sdílené klíče, jeden pro šifrování a druhý pro ochranu autentičnosti a integrity poslední výměny první fáze a pro celou druhou fázi. Ve třetí výměně účastníci posílají v zašifrované podobě

[3] ISAKMP (Internet Security Association and Key Management Protocol) poskytuje rámec pro správu klíče. Sám o sobě nediktuje konkrétní algoritmus výměny klíče, ale obsahuje definované výměny, payloady a postupy jejich zpracování. IKE je jedním z profilů tohoto rámce. IPSec DOI (Domain of Interpretation) instancuje ISAKMP pro použití s IP.

svou identitu spolu s důkazem této identity. Zároveň tím poskytují ochranu autentičnosti a integrity proběhlé DH výměně.

Phase 1 - Aggresive Mode Agresivní mód se skládá pouze ze tří zpráv, ale nechrání identity komunikujících a neumožňuje respondérovi vybrat si kryprografickou sadu. Iniciátor nabízí pouze jednu a respondér buď souhlasí, nebo ne. V první zprávě tedy iniciátor rovnou posílá svou DH hodnotu, svou identitu a návrh jedné konkrétní kryptosady. Pokud s ní respondér souhlasí, odvodí si sdílené klíče, pošle ve druhé zprávě tutéž kryptosadu, svou DH hodnotu, svou identitu a důkaz této identity. Nyní si iniciátor odvodí sdílené klíče a pošle ve třetí zprávě důkaz svojí identity.

Odvození klíčů pro IKE SA a důkaz identity Jak již bylo zmíněno, výstupem první fáze protokolu IKE je bezpečnostní asociace IKE SA, jejíž součástí je i odvození sdílených klíčů, které se použijí v posledních dvou zprávách první fáze a ve druhé fázi protokolu. Tyto klíče jsou narozdíl od klíčů pro protokoly AH a ESP obousměrné. Jejich výpočtu předchází odvození hodnoty SKEYID počítané použitím pseudonáhodné funkce s klíčem (prf). Konkrétní formule závisí na dohodnuté metodě ochrany autentičnosti a integrity. Při použití digitálních podpisů je SKEYID = $\mathrm{prf}(N_1\_i|N_1\_r, g^{xy})$, kde N_1_i a N_1_r jsou nonce iniciátora a respondéra, | značí spojení řetězců a g^{xy} je sdílené DH tajemství. Z hodnoty SKEYID jsou poté opět pomocí prf postupně odvozeny hodnoty:

$\mathrm{SKEYID\_d} = \mathrm{prf}(\mathrm{SKEYID}, g^{xy}|C_i|C_r|0)$,

$\mathrm{SKEYID\_a} = \mathrm{prf}(\mathrm{SKEYID}, \mathrm{SKEYID\_d}|g^{xy}|C_i|C_r|1)$,

$\mathrm{SKEYID\_e} = \mathrm{prf}(\mathrm{SKEYID}, \mathrm{SKEYID\_a}|g^{xy}|C_i|C_r|2)$,

kde C_i a C_r jsou cookie iniciátora a respondéra. Hodnota SKEYID_a je sdílený klíč pro ochranu autentičnosti a integrity a hodnota SKEYID_e sdílený klíč pro šifrování. Ostatní formule lze nalézt v [4]. Důkaz identity je proveden pomocí hodnot HASH_I pro iniciátora a HASH_R pro respondéra, opět vypočtených pomocí prf:

$\mathrm{HASH\_I} = \mathrm{prf}(\mathrm{SKEYID}, g^x|g^y|C_i|C_r|\mathrm{SA_i}|\mathrm{ID_i})$,

$\mathrm{HASH\_R} = \mathrm{prf}(\mathrm{SKEYID}, g^y|g^x|C_r|C_i|\mathrm{SA_i}|\mathrm{ID_r})$,

kde g^x je DH hodnota iniciátora, g^y je DH hodnota respondéra, $\mathrm{ID_i}$ je identita iniciátora, $\mathrm{ID_r}$ je identita respondéra a $\mathrm{SA_i}$ je návrh kryptografických sad poslaný iniciátorem (je to soubor všech navržených kryptosad, nejedná se o sadu, na které se oba účastníci dohodli). Pokud je zvolenou metodou ochrany autentičnosti a integrity digitální podpis, nejsou hodnoty HASH_I a HASH_R posílány přímo, ale pošlou se jejich podpisy.

Phase 2 - Quick Mode V této fázi protokolu dochází k dohodě na bezpečnostních parametrech a k odvození klíče pro protokol AH nebo ESP a může ji zahájit libovolný z účastníků. Skládá se ze tří zpráv zabezpečených pomocí klíčů SKEYID_a a SKEYID_e . V první zprávě iniciátor posílá hodnotu HASH(1), nabídku kryptografických sad a novou nonci N_2_i. Ve druhé zprávě respondér odpoví hodnotu HASH(2), kryptosadu, kterou si zvolil, a svou novou nonci N_2_r. Ve třetí zprávě iniciátor posílá hodnotu HASH(3) a potvrzuje tak dohodu. Jednotlivé položky zpráv jsou šifrované klíčem SKEYID_e. Hodnoty HASH zajišťují ochranu autentičnosti a integrity zpráv. Počítají se pomocí prf s klíčem SKEYID_a přes významné položky příslušné zprávy.[4] Oba účastníci nyní mohou odvodit sdílený klíčovací materiál KEYMAT:

$$\mathrm{KEYMAT} = \mathrm{prf}(\mathrm{SKEYID\_d}, \mathrm{protocol}|\mathrm{SPI}|N_1\_i|N_1\_r).$$

Položka protocol značí, pro který protokol bude klíč použit, a položka SPI[5] identifikuje, které SA bude klíč patřit. Je důležité upozornit, že v rámci jedné dohody každý účastník odvozuje pomocí KEYMAT dva různé klíče (tj. dvě různé SA), jeden pro příchozí a druhý pro odchozí zprávy komunikace. Klíč pro odchozí komunikaci je odvozen z KEYMAT počítaného pomocí SPI voleného protistranou, klíč pro příchozí komunikaci je odvozen z KEYMAT počítaného pomocí vlastního SPI.

Takto popsaný rychlý mód je tzv. základní a neposkytuje PFS (Perfect Forward Secrecy, viz 3.2). Pokud je PFS požadována, je součástí první zprávy nová DH hodnota iniciátora, která je navíc i dalším vstupem do HASH(1), a součástí druhé zprávy je nová DH hodnota respondéra, která je dalším vstupem do HASH(2). Nové DH tajemství se následně použije jako další vstup při odvození KEYMAT. Konkrétní formule a další podrobnosti jsou popsány v [4].

[4]Pro HASH(2) je součástí vstupu do prf navíc nonce iniciátora první fáze N_1_i a pro HASH(3) jsou součástí vstupu do prf nonce N_1_i a N_1_r.

[5]SPI se posílá v rámci popisu kryptografické sady (podrobněji v [9]).

Poznámka Pro IKEv1 existují dokonce čtyři možnosti zajištění ochrany autentičnosti a integrity (šifrování s veřejným klíčem lze použít v původní nebo v revidované verzi) a první fáze může být provedena ve dvou módech. Celkem tedy existuje osm variant IKEv1.

2.4.2 IKEv2

Nová verze protokolu IKE je popsána (na rozdíl od IKEv1) jediným dokumentem [6]. Vznikla jako zjednodušení a zúplnění původní verze, ale některé změny jsou tak zásadní, že není s původní verzí zpětně kompatibilní. Nejdůležitějšími rozdíly je právě snížení počtu dokumentů popisujících protokol a dále nahrazení osmi variant první fáze IKEv1 jedinou variantou úvodních výměn IKEv2 (různé metody ochrany autentičnosti a integrity ovlivní pouze výpočet jedné z položek). Další rozdíl je ten, že dohoda již neprobíhá ve fázích, ale v jednotlivých výměnách (tj. dvojicích zpráv (požadavek, odpověď)). Vždy platí, že další výměna nezačne, dokud není dokončena předchozí. První dvě výměny ustanoví bezpečnostní asociaci IKE SA a zároveň první bezpečnostní asociaci pro konkrétní protokol, zvanou Child SA. Poté mohou v libovolném počtu a pořadí následovat další výměny týkající se ustanovení nových SA (CREATE_CHILD_SA) nebo správy existujících SA. V určitých případech tedy k dohodě stačí pouze 4 zprávy, na rozdíl od minimálního počtu 6 zpráv v IKEv1 (v každé fázi 3).

Úvodní výměny IKEv2 Zmíněné první dvě výměny se nazývají úvodní. Musí proběhnout při každém zavolání IKEv2 v pevně daném pořadí. Jedná se o výměnu IKE_SA_INIT následovanou výměnou IKE_AUTH. V první zprávě iniciátor posílá nabídku jím podporovaných kryptografických algoritmů, svou DH hodnotu a nonci N_i. Respondér ve druhé zprávě posílá svou volbu kryptografického algoritmu, DH hodnotu a nonci N_r. V tomto momentě mohou oba účastníci vypočítat hodnotu SKEYSEED a z ní odvodit sedm tajných hodnot (SK_d, SK_ai, SK_ar, SK_ei, SK_er, SK_pi a SK_pr) následujícím způsobem:

$$\text{SKEYSEED} = \text{prf}(N_i|N_r, g^{xy}),$$

$$\{\text{SK\_d}|\text{SK\_ai}|\text{SK\_ar}|\text{SK\_ei}|\text{SK\_er}|\text{SK\_pi}|\text{SK\_pr}|\} = \text{prf+}(\text{SKEYSEED}, N_i|N_r|\text{SPIi}|\text{SPIr}),$$

kde pfr+ značí funkci, jejíž výstupem je pseudonáhodný proud tvořený zřetězenými výstupy funkcí prf. Konkrétní definici lze nalézt v [6]. Dvojice (SK_ei, SK_er) a (SK_ai, SK_ar) tvoří klíče pro IKE SA, kde klíče SK_e se použijí pro šifrování (zprávy se šifrují bez hlavičky) a klíče SK_a se použijí pro

ochranu autentičnosti a integrity zpráv posílaných v dalších výměnách. Dvojice (SK_pi, SK_pr) je použita v následující výměně při vytvoření položky AUTH[6] a hodnota SK_d je použita při odvozování klíčů pro Child SA. Ve třetí zprávě iniciátor pošle svou identitu a dokáže znalost tajemství odpovídající této identitě. Dále posílá položku AUTH, pomocí které zaručí ochranu autentičnosti a integrity první poslané zprávy. Následuje nabídka kryptografických algoritmů pro Child SA s selektory[7]. Respondér provede potřebná ověření a odešle čtvrtou zprávu. V té i on posílá a prokazuje svou identitu, zajistí ochranu autentičnosti a integrity druhé zprávy položkou AUTH a dokončí dohodu na Child SA volbou kryptografického algoritmu a souhlasem se selektory. Potřebná ověření nyní provede iniciátor. Po ověření výměny oba odvodí klíče pro Child SA. (Odvození proběhne stejně jako při výměně CREATE_CHILD_SA, podrobněji bude tedy popsáno v následujícím odstavci.)

CREATE_CHILD_SA Tato výměna se používá pro ustanovení nových Child SA a může ji zahájit libovolný z účastníků poté, co byly ukončeny úvodní výměny. Skládá se ze dvou zpráv, které jsou zabezpečeny klíči SK_e a SK_a. (Termín iniciátor bude tedy odpovídat iniciátorovi této výměny.) V první zprávě iniciátor posílá návrh kryptografických algoritmů, nonci N_i a selektory. Podobně jako ve druhé fázi IKEv1 může pro zajištění PFS zaslat novou DH hodnotu. Respondér ve druhé zprávě posílá zvolený kryptografický algoritmus, svou nonci N_r a případně svou DH hodnotu. Nyní oba účastníci odvodí klíčovací materiál KEYMAT:

$$\text{KEYMAT} = \text{prf+}(\text{SK\_d}, N_i|N_r).$$

Podobně jako v IKEv1, pokud je požadována PFS, jsou součástí výměny ještě nové DH hodnoty a vstupem do KEYMAT je navíc sdílené DH tajemství. Narozdíl od IKEv1 se KEYMAT nepočítá dvakrát, ale vypočítá se v takové délce, která je potřebná. Podrobnější informace lze nalézt v [6].

Poznámka Vidíme, že bezpečnost protokolu IPSec závisí v největší míře právě na zabezpečení dohody na klíčích pro protokoly AH a ESP. Ve zbylém textu se tedy zaměříme pouze na důkaz bezpečnosti protokolu IKE. Ten provedeme v rámci modelového prostředí, které popíšeme v následující kapitole.

[6]Konkrétní formule lze nalézt v [6].

[7]Selektory definují rozsahy zdrojových a cílových adres účastníků.

Kapitola 3

Modelové prostředí a základní pojmy

V této kapitole nejprve popíšeme obecný model protokolu výměny klíče v prostředí, kdy není na začátku protokolu známa identita protistrany, ale jednotliví účastníci ji odhalí až v průběhu protokolu (to odpovídá běžným případům v praxi, konkrétně protokolu IKE)[1]. Dále představíme model útočníka na tento protokol, včetně popisu a cíle tohoto útoku. Zároveň vysvětlíme použité pojmy. Na základě toho pak definujeme pojem bezpečnosti protokolu výměny klíče. Ve zbytku kapitoly se budeme zabývat konkrétním protokolem výměny klíče Σ_0, který reprezentuje zjednodušenou verzi protokolu výměny klíče IKE používajícího k ochraně autentičnosti a integrity digitální podpis. Pro tento protokol také později provedeme důkaz bezpečnosti.

3.1 Model protokolu výměny klíče

Protokol výměny klíče modelujeme jako pluralitní protokol (protokol s více účastníky), během nějž probíhá komunikace dvojic účastníků a jehož výstupem je tajný klíč. Každý účastník může mít v jednu chvíli spuštěnu jednu nebo více instancí protokolu. Základními požadavky na tento protokol jsou schopnost účastníků ověřit identitu protistrany, konzistence (pokud účastník A na konci protokolu ustanovil klíč k a věří, že ho sdílí s účastníkem B, pak pokud B ustanovil klíč k, musí věřit, že ho sdílí s A, a naopak.) a utajení klíče (žádný třetí účastník nesmí být schopen zjistit žádnou informaci o klíči).[2] Dále předpokládáme, že identity účastníků nejsou protistraně předem známy, ale jsou odhaleny až v průběhu protokolu (účastníci navazují komunikaci

[1] Jedná se o tzv. post-specified peer model, jak je popsán v [2].

[2] Ve všech těchto případech předpokládáme, že oba účastníci protokolu jsou čestní.

s konkrétní adresou). To umožňuje skrytí identit před pasivními útočníky a jednomu z účastníků i před těmi aktivními (tento účastník odhalí svou identitu až po ověření identity protistrany).

Sezení Každou instanci protokolu budeme nazývat *sezení (session)*. Sezení je u účastníka lokálně spuštěno poté, co je *aktivován* (dostane podnět k zahájení sezení), a to buď jako *iniciátor* (požadavek na zahájení protokolu přichází přímo od účastníka, např. je iniciován nějakou aplikací), nebo jako *respondér* (požadavek přichází z vnějšku, od nějakého jiného účastníka).

Lokálně je sezení u daného účastníka aktivováno vstupní trojicí (P, s, d), kde P je identita tohoto účastníka, s je identifikátor sezení (*sessionID*) a d je adresa protistrany. Pro protistranu se běžně používá termín *peer*. Identifikátor s je volen tak, aby byl unikátní mezi všemi identifikátory, které účastník P používá (sezení mohou probíhat souběžně a s určuje, kterému sezení konkrétní zpráva patří). Globálním identifikátorem tohoto sezení je dvojice (P, s).

V rámci sezení jsou vytvářeny odchozí zprávy a jsou zpracovávány zprávy příchozí. Každý z účastníků si udržuje lokální stav příslušející tomuto sezení, který po ukončení sezení vymaže. Navíc může účastník udržovat dodatečný stav (obsahující dlouhodobá tajemství, jako je podpisový klíč), ke kterému přistupují různá sezení a který není součástí žádného lokálního stavu.

Proběhlo-li sezení v pořádku, jeho výstupem je veřejná trojice (P, s, Q), kde Q je identita protistrany, a dále tajná hodnota sdíleného tajemství, které budeme nazývat *session key*. Takové sezení se nazývá *dokončené (completed)* (může se stát, že sezení je dokončené u jedné z protistran, ale ne u druhé).

Pokud je z nějakého důvodu sezení ukončeno předčasně a nedojde k ustanovení session key, nazývá se *přerušené (aborted)*. Výstupem takového sezení je speciální symbol značící neúspěch dohody.

Od chvíle, kdy je o session key dohodnutém dokončeným sezením rozhodnuto, že se již nebude používat a bude vymazán (spojení, které ho využívalo ke svému zabezpečení, je ukončeno), se toto sezení nazývá *vypršelé (expired)*.

Zbývá ještě definovat pojem shodné sezení, který je pro popis bezpečnosti protokolu velice důležitý.

Definice 1 (Shodné sezení) *Buď (P, s) dokončené sezení s veřejným výstupem (P, s, Q). Sezení (Q, s) se nazývá shodné sezení se sezením (P, s), jestliže platí jedna z následujících možností:*

1. *(Q, s) není dokončené;*
2. *(Q, s) je dokončené s veřejným výstupem (Q, s, P).*

Poznámka Shodné sezení je definováno pouze pro dokončená sezení.

3.2 Model útočníka

Dalším důležitým prvkem pro popis bezpečnosti protokolu je model útočníka[3] na tento protokol. Útočníkem je pravděpodobnostní polynomiální stroj. Předpokládáme, že má plnou kontrolu nad komunikačními linkami, tedy může odposlouchávat, zpožďovat, zahazovat a měnit všechny přenášené zprávy a může také vkládat jím generované zprávy. Zároveň rozhoduje o aktivaci účastníka a vidí všechny veřejné výstupy sezení.

Navíc k těmto schopnostem může útočník získat tajné informace uložené v paměti účastníků pomocí následujících tří útoků, které se souhrnně označují jako *session exposure (odhalení sezení)*:

- *session state reveal* - útočník zjistí lokální stav nějakého *nedokončeného* sezení;
- *session key querry* - útočník zjistí tajný session key nějakého *dokončeného* sezení;
- *party corruption* - útočník zjistí všechny informace v paměti nějakého účastníka (včetně dlouhodobých tajemství); od chvíle, kdy je účastník „zkorumpován“, je plně ovládán útočníkem.

Sezení, na které byl použit některý z těchto útoků, nazýváme *odhalené (exposed)*. Pokud sezení vyprší, útočník na něj již nesmí zaútočit. Stále ale může zkorumpovat účastníka tohoto sezení, a to i v případě, že s ním shodné sezení ještě nevypršelo, nebo dokonce ani nebylo dokončeno.

Perfektní dopředná bezpečnost (Perfect Forward Secrecy - PFS) Pokud protokol zajišťuje ochranu vypršelých sezení i v případě zkorumpování účastníka, zajišťuje perfektní dopřednou bezpečnost.

Definice 2 (Perfektní dopředná bezpečnost) *Poté, co jsou smazány tajné klíče příslušející určitému spojení mezi dvěma účastníky, žádný útočník není (bez použití útoku hrubou silou na prostor všech session key) schopen zrekonstruovat tyto klíče ani v případě, že zaznamenal všechna přenášená data a má přístup k dlouhodobým tajemstvím obou účastníků.*

Útok na protokol výměny klíče Při útoku na protokol výměny klíče je úkolem útočníka efektivně odlišit skutečnou hodnotu session key od náhodné hodnoty nezávislé na session key. Toto je formalizováno pomocí pojmu *testovací sezení (test session)*, které útočník volí libovolně ze všech dokončených, neodhalených a nevypršelých) sezení protokolu. Útok probíhá ve třech fázích:

[3]Jedná se o tzv. UM (Unauthenticated Links Model)

1. Na začátku útoku je nejprve náhodně zvolena hodnota bitu b. Poté útočník zvolí testovací sezení a je mu poskytnuta hodnota v (*challenge session key*), která v případě $b = 0$ je skutečnou hodnotou session key zvoleného sezení, a v případě $b = 1$ je náhodnou hodnotou stejné délky a se stejným rozdělením, ale nezávislou na session key.

2. Útočník nyní provádí obvyklé akce proti protokolu, nesmí ale použít útoky session exposure na testovací ani s ním shodné sezení.

3. Na konci svého běhu odpoví bit b', který je jeho odhadem hodnoty bitu b.

Útočník při útoku uspěl, pokud $b' = b$. Tohoto útočníka budeme nazývat *session key útočník (SK-útočník).*

3.3 Bezpečnost protokolu výměny klíče

Poté, co jsme popsali model protokolu výměny klíče spolu se schopnostmi útočníka a popsali jsme, jakým způsobem útok probíhá, můžeme na základě toho definovat bezpečnost tohoto protokolu, kterou budeme nazývat *session key bezpečnost (SK-bezpečnost).* Neformálně můžeme říci, že SK-bezpečnost znamená, že SK-útočník interakcí s protokolem ani pomocí útoku na jiná sezení a účastníky nezjistí nic o hodnotě session-key. Formální definice následuje.

Definice 3 (SK-bezpečnost) *Protokol výměny klíče π se nazývá SK-bezpečný, jestliže pro každého SK-útočníka $\mathcal{A}$ útočícího na π platí:*

1. *Pokud dva nezkorumpovaní účastníci dokončí shodná sezení v běhu protokolu π při útoku útočníka $\mathcal{A}$, pak až na zanedbatelnou pravděpodobnost je výstup session key těchto sezení shodný.*

2. *Útočník $\mathcal{A}$ uspěje při útoku (na testovací sezení) s pravděpodobností nejvýše $\frac{1}{2} + \epsilon$ pro ϵ zanedbatelné.*

Důkaz bezpečnosti protokolu se provádí na základě generických vlastností v protokolu použitých primitivů, a nebude tedy závislý na konkrétních kryptografických algoritmech. Protokol je v popsaném modelu bezpečný, pokud se ukáže, že libovolnou akci útočníka, která porušuje některou z požadovaných bezpečnostních vlastností protokolu, lze převést na explicitní algoritmus, který prolomí bezpečnost některého z primitivů. Dokud tedy budou bezpečné primitivy, protokol bude také bezpečný.

Kapitola 4

Protokol Σ_0

Nyní si představíme konkrétní protokol výměny klíče Σ_0, který reprezentuje zjednodušenou verzi protokolu IKE používající k ochraně autentičnosti a integrity digitální podpis. Tento protokol obsahuje zásadní kryptografické prvky a vlastnosti plnohodnotného protokolu IKE a lze na něm srozumitelně ukázat důkaz SK-bezpečnosti. Σ_0 patří do rodiny protokolů SIGMA (jedná se o protokoly výměny klíče, které pomocí kombinace podpisového schématu a schématu MAC zajišťují ochranu autentičnosti a integrity Diffie-Hellmanově výměně; podrobný popis v [8]). Protokol, jak ho popíšeme, poskytuje perfektní dopřednou bezpečnost.

Dále se pokusíme přiblížit, jak se dokazuje SK-bezpečnost tohoto protokolu. Důkaz (jak je uveden v [2]) opravdu popíšeme pouze zhruba a podrobně se budeme zabývat až jeho verzí z pohledu konkrétní bezpečnosti. Podrobně ale popíšeme simulátory, které jsou pro důkaz klíčové.

4.1 Popis protokolu

Předpokládáme, že oba účastníci mají informace o použité grupě (my zde uvedeme jako příklad podgrupu grupy $\mathbb{Z}_p^*$ řádu q, kde p a q jsou prvočísla) a o konkrétních primitivech použitých k zajištění bezpečnosti (podpisové schéma, shéma MAC a rodina pseudonáhodných funkcí), které během protokolu použijí, a také že mají k dispozici veřejné podpisové klíče protistrany. V praxi je dohoda na konkrétních algoritmech a grupě součástí výměny a je součástí dat, jimž je poskytnuta ochrana autentičnosti a integrity. O použité grupě zároveň předpokládáme, že v ní platí předpoklad o nerozlišitelnosti DH tajemství od náhodné hodnoty (DDH-předpoklad), a že použité primitivy jsou bezpečné. (Podrobné definice DDH-předpokladu a jednotlivých primitivů včetně jejich bezpečnosti budou uvedeny v kontextu konkrétní bezpečnosti.)

V rámci protokolu budeme rozlišovat tři typy zpráv, *úvodní (start message)*, *odpovědní (response message)* a *závěrečnou (finish message)*. Každý účastník vždy zpracuje pouze první příchozí zprávu daného typu příslušející jednomu konkrétnímu sezení (to je identifikováno hodnotou sessionID), ostatní zprávy stejného typu a se stejným identifikátorem budou ignorovány. Protokol nespecifikuje, kdo přesně zprávu dostane a kdy. Toto je řízeno útočníkem. Jednotlivé zprávy budou odesílány na nějakou (konkrétní) adresu, která není nijak logicky spojena s identitou účastníka.

SessionID zde má nejen funkci identifikátoru sezení, ale slouží zároveň jako záruka čerstvosti zprávy. V praxi se za tímto účelem používají např. nonce. Zde volí celý sessionID iniciátor a respondér musí ověřit, že je pro něj jednoznačný. V praxi volí každý svou unikátní hodnotu a sessionID vznikne jejich zřetězením. V tom případě obsahuje první zpráva iniciátora pouze první část sessionID. Další zprávy pak již obsahují sessionID celé.

Značení Parametry použité grupy jsou prvočísla p, q, kde q dělí $p-1$, a prvek $g \in \mathbb{Z}_p^*$ řádu q. Hodnoty x, y značí DH exponenty iniciátora a respondéra a g^x, g^y značí jejich DH hodnoty. Hodnota g^{xy} značí sdílené DH tajemství. Operace $\hat{}$ je modulární umocňování, jak je definováno pro danou grupu. Položky ID_i a ID_r značí reálné identity obou účastníků. Zkratky SIG, MAC a PRF značí po řadě podpisové schéma, schéma MAC a rodinu pseudonáhodných funkcí. Indexy i a r pro schéma SIG značí soukromé podpisové klíče iniciátora a respondéra. Indexy pro schémata MAC a PRF značí vstupní klíče. Hodnota k_0 je výsledný session key. Příznaky '0' a '1' rozlišují, zda informace sloužící k ochraně autentičnosti a integrity byla vytvořena v roli iniciátora (0) nebo v roli respondéra (1). Ty nejsou nutnou součástí protokolu Σ_0. Zde slouží ke zjednodušení důkazu, bezpečnost platí i bez nich (argumentaci lze nalézt v [2]).

Poznámka Následuje schéma popisující jednotlivé akce účastníků během protokolu. Připomeňme, že každý účastník zná na začátku protokolu parametry p, q a g, konkrétní algoritmy pro vytváření digitálních podpisů a MACů a konkrétní pseudonáhodnou funkci. Dále má každý účastník k dispozici svůj soukromý podpisový klíč a veřejný klíč příslušející soukromému klíči protistrany.

Iniciátor (ID_i) **Respondér (ID_r)**

ID_i je aktivován jako iniciátor sezení,
zvolí unikátní sessionID s,
zahájí sezení (ID_i, s):
zvolí x náhodně ze $\mathbb{Z}_q$, spočte g^x,
x uloží do lokálního stavu,
odešle úvodní zprávu:
s, g^x

ID_r je úvodní zprávou aktivován jako respondér,
ověří unikátnost s a zahájí sezení (ID_r, s):
zvolí y náhodně ze $\mathbb{Z}_q$,
spočte: g^y, $\mathrm{SIG}_r(\text{`1'}, s, g^x, g^y)$, $g^{xy} = (g^x)^y$,
$k_1 = \mathrm{PRF}_{g^{xy}}(1)$, $k_0 = \mathrm{PRF}_{g^{xy}}(0)$,
$\mathrm{MAC}_{k_1}(\text{`1'}, s, \mathrm{ID}_r)$,
k_1, k_0 uloží do lokálního stavu, smaže y a g^{xy},
odešle odpovědní zprávu:
$s, g^y, \mathrm{ID}_r, \mathrm{SIG}_r(\text{`1'}, s, g^x, g^y), \mathrm{MAC}_{k_1}(\text{`1'}, s, \mathrm{ID}_r)$

přijme odpovědní zprávu se sessionID s,
veřejným klíčem účastníka ID_r ověří SIG_r,
spočte $g^{xy} = (g^y)^x$, $k_1 = \mathrm{PRF}_{g^{xy}}(1)$,
ověří obdržený MAC_{k_1}
<u>libovolné ověření selže:</u>
vymaže lokální stav,
výstup „(ID_i, s) přerušeno“
<u>obě ověření v pořádku:</u>
spočte $k_0 = \mathrm{PRF}_{g^{xy}}(0)$,
pošle závěrečnou zprávu:
$s, g^y, \mathrm{ID}_i, \mathrm{SIG}_i(\text{`0'}, s, g^y, g^x), \mathrm{MAC}_{k_1}(\text{`0'}, s, \mathrm{ID}_i)$,
vymaže lokální stav,
dokončí sezení s výstupem $(\mathrm{ID}_i, s, \mathrm{ID}_r)$, k_0

přijme závěrečnou zprávu se sessionID s,
veřejným klíčem účastníka ID_i ověří SIG_i,
ověří obdržený MAC_{k_1}
<u>libovolné ověření selže:</u>
vymaže lokální stav,
výstup „(ID_r, s) přerušeno“
<u>obě ověření v pořádku:</u>
dokončí sezení s výstupem $(\mathrm{ID}_r, s, \mathrm{ID}_i)$, k_0

Poznámka Takto popsaný protokol můžeme srovnat s popisem protokolu IKE. Vidíme, že se vlastně jedná o druhou a třetí výměnu hlavní fáze protokolu. První výměna ale obsahuje pouze dohodu na použitých kryptografických sadách, tedy její vynechání nevadí. Ostatně takto funguje agresivní m=od IKE. Existuje i verze protokolu, ve které si účastníci vymění celkem čtyři zprávy. Takto se dá zajistit aktivní ochrana identity pro respondéra sezení.

4.2 Idea důkazu SK-bezpečnosti protokolu Σ_0

V této části kapitoly představíme hlavní strukturu důkazu SK-bezpečnosti protokolu Σ_0. Jak již bylo řečeno, podrobný popis důkazu provedeme až z pohledu konkrétní bezpečnosti. Důkaz z pohledu asymptotické bezpečnosti používá stejnou argumentaci, přístup konkrétní bezpečnosti navíc měří míru bezpečnosti pomocí explicitní formule.

Abychom dokázali SK-bezpečnost protokolu Σ_0, budeme muset dokázat oba body definice 3. Tedy že pokud nezkorumpovaní účastníci ID_i a ID_r dokončí v běhu protokolu Σ_0 shodná sezení $(\mathrm{ID}_i, s, \mathrm{ID}_r)$ a $(\mathrm{ID}_r, s, \mathrm{ID}_i)$, pak až na zanedbatelnou pravděpodobnost je tajný výstup session key z obou těchto sezení shodný, a že neexistuje SK-útočník na Σ_0, který při svém útoku uspěje s nezanedbatelnou pravděpodobností.

První bod definice se dokáže snadno. Ukáže se, že oba vypočítají stejné DH tajemství g^{xy}, protože obě DH hodnoty jsou chráněny podpisovým schématem SIG. Aby mohl útočník DH hodnotu protistrany podvrhnout, musel by být schopen padělat příslušný podpis. Protože ale předpokládáme, že SIG je bezpečné, toto se může podařit pouze se zanedbatelnou pravděpodobností.

Důkaz druhého bodu definice je značně složitější. Cílem je ukázat, že útočník nedokáže efektivně rozlišit mezi skutečným session key a náhodnou hodnotou. Jinými slovy, že rozdíl mezi pravděpodobnostmi, že uhádne bit b a že neuhádne bit b, je zanedbatelný. Obecná metoda takového důkazu je ukázat, že pokud by existoval útočník, který tyto hodnoty dokáže efektivně rozlišit, pak by musel existovat útočník, který by dokázal efektivně útočit na některý z primitivů použitých v protokolu nebo na DDH-předpoklad použité grupy.

Značení útočníků SK-útočníka na Σ_0 budeme značit $\mathcal{A}$, útočníka na DDH-předpoklad (DDH-distinguisher) budeme značit $\mathcal{D}_{\mathrm{DDH}}$, útočníka na schéma MAC (MAC-forger) budeme značit $\mathcal{F}_{\mathrm{MAC}}$, útočníka na PRF (PRF-distinguisher) budeme značit $\mathcal{D}_{\mathrm{PRF}}$ a útočníka na schéma SIG (SIG-forger) budeme značit $\mathcal{F}_{\mathrm{SIG}}$. Definice jednotlivých útočníků budou uvedeny v kontextu konkrétní bezpečnosti.

Postup důkazu druhého bodu definice Hlavním cílem důkazu je ukázat, že libovolný útočník $\mathcal{A}$, který uspěje při rozlišení skutečného session key od náhodné hodnoty, může být použit k sestrojení útočníka $\mathcal{D}_{\mathrm{DDH}}$, který rozliší trojici (g^x, g^y, g^{xy}) od náhodné trojice (g^x, g^y, g^r) se stejnou advantage[1] jako $\mathcal{A}$, nebo lze sestrojit útočníka, který úspěšně útočí na jeden z primitivů.

Útočník $\mathcal{D}_{\mathrm{DDH}}$ tedy dostane vstupní trojici (g^x, g^y, z) a má určit, zda z je g^{xy} nebo g^r. Pokud by měl k dispozici útočníka $\mathcal{A}$, který dokáže rozlišit mezi skutečným session key a náhodnou hodnotou, mohl by se mu pokusit „podstrčit“ hodnoty ze vstupní trojice do testovacího sezení a odpovídat podle něj. Stačilo by v testovacím sezení použít hodnoty g^x a g^y jako DH hodnoty účastníků tohoto sezení a hodnotu z jako jimi odvozené DH tajemství. Jako challenge session key by útočníkovi poskytl hodnotu k_0 počítanou s použitím hodnoty z.

Protože ale $\mathcal{A}$ volí testovací sezení z dokončených sezení, musel by $\mathcal{D}_{\mathrm{DDH}}$ zahrnout své vstupní hodnoty do protokolu ještě před tím, než útočník volí testovací sezení. Musel by tedy uhodnout sezení, které si $\mathcal{A}$ vybere jako testovací. Již by nezáleželo na tom, zda se bude jednat o sezení, které dokončil účastník v roli iniciátora, nebo s ním shodné, které dokončil účastník v roli respondéra (obě budou počítat session key s použitím z). Dalším problémem je, že $\mathcal{D}_{\mathrm{DDH}}$ by svou akcí zasáhl do normálního chování obou peerů tohoto sezení, tedy bude to muset udělat tak, aby $\mathcal{A}$ neovlivnil.

Aby se ukázalo, že i přes zmíněné problémy lze úspěšného útočníka $\mathcal{A}$ využít, bude zavedena posloupnost simulátorů, které budou útočníkovi $\mathcal{A}$ vytvářet dojem, že provádí útok na Σ_0, a zároveň ovlivní zprávy posílané v rámci toho sezení, které hádají, že bude testovací. Jeden z těchto simulátorů bude představovat situaci, kdy ovlivněné sezení vydá skutečný session key (počítaný podle protokolu), a jeden bude představovat situaci, kdy vydá místo session key náhodnou hodnotu. Pokud ovlivněné sezení bude testovací (simulátor ho uhádl), v první simulaci $\mathcal{A}$ dostane jako challenge skutečný session key a ve druhé dostane jako challenge náhodnou hodnotu.

4.3 Simulátory

Nejprve definujeme, jak funguje obecný simulátor, který budeme značit $\mathcal{S}$. Na základě tohoto simulátoru pak definujeme simulátor $\hat{\mathcal{S}}$ a jeho konkrétní varianty, které jsou využity v důkazu. Pro každou z těchto variant bude

[1] Pro útočníka, jehož cílem je rozlišit skutečnou odpověď daného schématu od náhodné odpovědi, je jeho advantage rovna rozdílu pravděpodobnosti, že odpoví „náhodná hodnota“ v případě, že se skutečně jedná o náhodnou hodnotu, a pravděpodobnosti, že odpoví „náhodná hodnota“ v případě, že se jedná o skutečnou hodnotu.

$\hat{\mathcal{S}}(\mathcal{A})$ představovat pravděpodobnostní rozdělení běhů simulátoru $\hat{\mathcal{S}}$ během interakce s útočníkem $\mathcal{A}$.

4.3.1 Simulátor $\mathcal{S}$

$\mathcal{S}$ danému útočníkovi $\mathcal{A}$ simuluje běh protokolu Σ_0. Na začátku svého běhu pro parametry n (počet účastníků protokolu) a κ (bezpečnostní parametr) vytvoří pro každého z účastníků inicializační informace (dvojice podpisových klíčů). Poté spustí běh útočníka $\mathcal{A}$, který řídí aktivaci sezení mezi účastníky.

Po každé takové aktivaci simulátor provádí akce protokolu tak, jak by je dělali skuteční účastníci, a poskytuje útočníkovi zprávy, které by účastníci posílali. Také pokud útočník provede některý z útoků session exposure, předá mu simulátor příslušné informace. V případě zkorumpování nějakého účastníka simulátor po odevzdání všech jeho interních informací přestane účastníka provozovat. Když útočník položí dotaz na testovací sezení, poskytne mu simulátor session key tohoto sezení. Když útočník skončí a vydá svůj odhad, simulátor také skončí a se stejnou odpovědí.

4.3.2 Simulátor $\hat{\mathcal{S}}$

Na začátku svého běhu $\hat{\mathcal{S}}$ zvolí náhodně číslo T z množiny $\{1, \ldots, m\}$, kde m je apriorní horní mez počtu sezení, která $\mathcal{A}$ během svého běhu s n účastníky a s bezpečnostním parametrem κ iniciuje (aktivuje iniciátora sezení). Dále zvolí identitu R_0 náhodně z identit všech účastníků, náhodné prvky $x, y \in \mathbb{Z}_q$ a hodnoty k_1 a k_0 délky výstupu funkcí z PRF. Volba čísla T a identity R_0 vyjadřuje pokus simulátoru uhádnout, které sezení si $\mathcal{A}$ zvolí jako testovací a který z účastníků bude jeho respondérem.

V dalším kroku simulátor spustí útočníka $\mathcal{A}$ a provádí simulaci stejně jako $\mathcal{S}$ až na následující situace: 1) když provádí akce spojené s T-tým sezením (budeme ho značit (I_0, s_0)), které $\mathcal{A}$ iniciuje; 2) když nastane některá z *událostí přerušení* (popsány níže). Když nedojde k přerušení a $\mathcal{A}$ dokončí svůj běh, dokončí běh i $\hat{\mathcal{S}}$ a odpoví stejný bit.

Akce simulátoru spojené s T-tým sezením Následující akce se budou provádět, dokud nenastane některá z událostí přerušení. Použijí se hodnoty x, y, k_1 a k_0 volené v úvodu běhu simulátoru.

Úvodní zprávu sezení (I_0, s_0) vygeneruje simulátor pomocí hodnoty x (použije ji jako DH exponent iniciátora). Pokud je útočníkem $\mathcal{A}$ aktivováno sezení (R_0, s_0), kde R_0 je v roli respondéra, simulátor vytvoří odpovědní zprávu tohoto sezení s použitím hodnot y (DH exponent respondéra) a k_1 (klíč pro

MAC). Pokud sezení (I_0, s_0) přijme odpovědní zprávu, použije simulátor hodnotu k_1 k ověření přijatého MACu a k vytvoření MACu do závěrečné zprávy. Pokud sezení (R_0, s_0) obdrží závěrečnou zprávu, také k ověření MACu použije k_1. Pokud se sezení (I_0, s_0) dokončí, bude hodnotou session key k_0. Stejně tomu bude, pokud se dokončí sezení (R_0, s_0). Pokud pak $\mathcal{A}$ zvolí (I_0, s_0) nebo (R_0, s_0) jako testovací sezení, poskytne mu $\hat{\mathcal{S}}$ jako challenge hodnotu k_0.

Události přerušení Při události přerušení simulátor dále nepokračuje v simulaci, ale skončí svůj běh a odpoví bit 0. Jedná se o následující události:

- útočník před dokončením sezení (I_0, s_0) zkorumpuje I_0 nebo R_0;
- útočník provede proti (I_0, s_0) nebo (R_0, s_0) útok session state reveal;
- sezení (R_0, s_0), kde R_0 je v roli respondéra, je iniciováno dříve, než (I_0, s_0) odešle úvodní zprávu, nebo je iniciováno poté, ale přijatá úvodní zpráva obsahuje jinou DH hodnotu, než kterou sezení (I_0, s_0) odeslalo;
- (I_0, s_0) obdrží odpovědní zprávu dřív, než bylo aktivováno (R_0, s_0), kde R_0 je v roli respondéra, nebo ji obdrží poté, ale ta obsahuje jinou DH hodnotu, než kterou sezení (R_0, s_0) odeslalo;
- (I_0, s_0) se přeruší;
- útočník zvolí jiné testovací sezení než (I_0, s_0) nebo (R_0, s_0), nebo jedno z nich zvolí, ale to je dokončeno s jiným peerem než R_0 nebo I_0;
- útočník dokončí útok, aniž by zvolil testovací sezení, nebo ukončí běh předtím, než inicioval T sezení.

Poznámka Události přerušení slouží k tomu, aby simulátor nepokračoval v simulaci, pokud neuhodl testovací sezení a jeho respondéra. Zároveň díky tomu simulátor nečeká, až $\mathcal{A}$ testovací sezení zvolí, ale pokud je jasné, že T-té sezení již nemůže splňovat podmínky testovacího sezení, je simulace přerušena. Z toho tedy vyplývá, že $\hat{\mathcal{S}}$ odpovídá podle výstupu $\mathcal{A}$ pouze v situaci, kdy uhodl testovací sezení a jeho respondéra. K uhádnutí dojde, pokud útočník zvolí jako testovací sezení buď (I_0, s_0) nebo (R_0, s_0).

Jednotlivé verze $\hat{\mathcal{S}}$ Jednotlivé verze $\hat{\mathcal{S}}$ se od sebe budou lišit pouze způsobem, jakým daný simulátor volí v úvodním kroku simulace hodnoty k_1 a k_0. Budeme je podle toho také označovat. Symbol $r()$ reprezentuje náhodnou a nezávislou volbu řetězce vhodné délky. Verze $\hat{\mathcal{S}}$ jsou následující:

- $\hat{\mathcal{S}}-$REAL: $k_0 \leftarrow \mathrm{PRF}_{g^{xy}}(0)$, $k_1 \leftarrow \mathrm{PRF}_{g^{xy}}(1)$
- $\hat{\mathcal{S}}-$RPRF: $k_0 \leftarrow \mathrm{PRF}_{k}(0)$, $k_1 \leftarrow \mathrm{PRF}_{k}(1)$, $k \leftarrow r()$
- $\hat{\mathcal{S}}-$ALLR: $k_0 \leftarrow r()$, $k_1 \leftarrow r()$
- $\hat{\mathcal{S}}-$HYBR: $k_0 \leftarrow r()$, $k_1 \leftarrow \mathrm{PRF}_{k}(1)$, $k \leftarrow r()$
- $\hat{\mathcal{S}}-$RAND: $k_0 \leftarrow r()$, $k_1 \leftarrow \mathrm{PRF}_{g^{xy}}(1)$

Kapitola 5

Definice základních pojmů v kontextu konkrétní bezpečnosti

V této kapitole uvedeme znovu definice základních pojmů bezpečnosti schémat výměny klíče, a to definici SK-útočníka a definici SK-bezpečnosti, tentokrát však v kontextu konkrétní bezpečnosti. Dále uvedeme slíbené definice použitých kryptografických primitivů SIG, MAC a PRF a také předpokladu DDH. Na závěr ještě definujeme další pojmy později použité v důkazu.

5.1 SK-útočník a SK-bezpečnost

SK-útočník Nechť $\mathcal{A}$ je útočník, který útočí na protokol výměny klíče π a který má k dispozici orákulum $\mathcal{O}^{\text{state}}(\cdot)$, které pro dané nedokončené sezení vrací vnitřní stav tohoto sezení, orákulum $\mathcal{O}^{\text{sk}}(\cdot)$, které pro dané dokončené sezení (které nevypršelo) vrací hodnotu session key tohoto sezení, orákulum $\mathcal{O}^{\text{corr}}(\cdot)$, které pro daného účastníka vrací celý obsah jeho paměti, orákulum $\mathcal{O}^{\text{resp}}(\cdot,\cdot)$, které pro daného účastníka a danou úvodní zprávu vrací odpovědní zprávu daného účastníka, a oráklulum $\mathcal{O}^{\text{test}}(\cdot)$, které pro dané dokončené a neodhalené sezení (testovací) vrací challenge session key, který je náhodně volen buď jako skutečný session key daného sezení, nebo jako náhodná a na session key nezávislá hodnota stejné délky a se stejným rozdělením, jako má skutečný session key.

Dále nechť pro tohoto útočníka platí, že čas jeho běhu nepřekročí t, počet jeho dotazů na $\mathcal{O}^{\text{state}}(\cdot)$ nepřekročí q_{state}, počet jeho dotazů na $\mathcal{O}^{\text{sk}}(\cdot)$ nepřekročí q_{sk}, počet jeho dotazů na $\mathcal{O}^{\text{corr}}(\cdot)$ nepřekročí q_{corr} a počet jeho dotazů na $\mathcal{O}^{\text{resp}}(\cdot,\cdot)$ nepřekročí q_{resp}.

Útok výše popsaného útočníka $\mathcal{A}$ útočícího na testovací sezení protokolu π popíšeme pomocí experimentu $Exp^{\text{test}}_{\pi,\mathcal{A}}$:

- na začátku experimentu je provedena náhodná volba $b \xleftarrow{R} \{0,1\}$;
- útočník $\mathcal{A}$ z množiny všech dokončených sezení protokolu π zvolí testovací sezení (P_0, s_0) a pošle ho jako dotaz na orákulum $\mathcal{O}^{\text{test}}(\cdot)$. Orákulum odpoví hodnotou v (challenge session key): pokud $b = 0$, v je skutečná hodnota session key sezení (P_0, s_0), pokud $b = 1$, v je náhodná na session key nezávislá hodnota se stejným rozdělením a stejné délky jako skutečný session key sezení (P_0, s_0);
- útočník může pokračovat běžnými akcemi proti protokolu, tj. může posílat dotazy na orákulum $\mathcal{O}^{\text{state}}(\cdot)$ tvaru (P, s), kde (P, s) je nedokončené sezení, může posílat dotazy na orákulum $\mathcal{O}^{\text{sk}}(\cdot)$ tvaru (P, s), kde (P, s) je dokončené sezení, různé od sezení (P_0, s_0) a sezení (ID, s_0) s ním shodného, může posílat dotazy na orákulum $\mathcal{O}^{\text{corr}}(\cdot)$ tvaru P, kde P je různé od P_0 a od identity respondéra ID, a může posílat dotazy na orákulum $\mathcal{O}^{\text{resp}}(\cdot, \cdot)$ tvaru $(P, (s, g^x))$;
- na konci svého běhu odpoví $\mathcal{A}$ hodnotu b', což je jeho odhad na bit b.

Řekneme, že útočník $\mathcal{A}$ je při útoku na π úspěšný, pokud platí $b = b'$. Advantage útočníka $\mathcal{A}$ při útoku na testovací sezení protokolu π vyjádříme jako:

$$\mathbf{Adv}_{\pi}^{\text{test}}(\mathcal{A}) = |\Pr[\mathcal{A} \to 1 | b = 0] - \Pr[\mathcal{A} \to 1 | b = 1]|$$

Takto popsaného útočníka $\mathcal{A}$ nazýváme $(t, q_{\text{state}}, q_{\text{sk}}, q_{\text{corr}}, q_{\text{resp}})$-SK-útočník.

Poznámka Pro jednoduchost budeme v dalším textu používat zkrácené značení q pro $q_{\text{state}}, q_{\text{sk}}, q_{\text{corr}}, q_{\text{resp}}$.

Definice 4 ((t, q, δ, ϵ)-SK-bezpečnost) *Protokol výměny klíče π se nazývá (t, q, δ, ϵ)-SK-bezpečný, jestliže pro každého (t, q)-SK-útočníka $\mathcal{A}$ útočícího na π platí:*

1. *Pokud dva nezkorumpovaní účastníci dokončí shodná sezení v běhu protokolu π při útoku útočníka $\mathcal{A}$, pak pravděpodobnost, že je výstup session key těchto sezení shodný, je větší než $1 - \delta$.*
2. *Útočník $\mathcal{A}$ při svém útoku na testovací sezení uspěje s pravděpodobností menší než $\frac{1}{2} + \epsilon$, neboli $\mathbf{Adv}_{\pi}^{\text{test}}(\mathcal{A}) < \epsilon$.*

5.2 DDH-předpoklad a primitivy PRF, SIG a MAC

V této části kapitoly definujeme v kontextu konkrétní bezpečnosti základní stavební kameny protokolu Σ_0. Patří sem předpoklad DDH (Decisional Diffie-Hellman), rodina pseudonáhodných funkcí PRF, podpisové schéma SIG a schéma MAC.

Poznámka Předpoklad DDH uvedeme pro podgrupu grupy $\mathbb{Z}_p^*$, kterou jsme použili v popisu protokolu Σ_0. Předpoklad lze podobně formulovat i pro ostatní grupy (např. grupy na eliptických křivkách).

(t, ϵ)-DDH-předpoklad Nechť κ je bezpečnostní parametr, nechť p a q jsou prvočísla taková, že q je délky κ a q dělí $p-1$, a nechť g je řádu q v $\mathbb{Z}_p^*$. Definujeme rozdělení Q_0 a Q_1 následovně:

$$Q_0 = \left\{(p, g, g^x, g^y, g^{xy}) : x, y \xleftarrow{R} \mathbb{Z}_q\right\}, Q_1 = \left\{(p, g, g^x, g^y, g^r) : x, y, r \xleftarrow{R} \mathbb{Z}_q\right\}$$

Pak pro každý distinguisher $\mathcal{D}$, který je omezený výpočetním časem t platí, že:

$$\left|\Pr[\mathcal{D}^{Q_0} \to 1] - \Pr[\mathcal{D}^{Q_1} \to 1]\right| < \epsilon$$

Rodina funkcí PRF Buď $\mathcal{F}^0 : Keys(\mathcal{F}^0) \times \{0,1\}^l \to \{0,1\}^L$ rodina funkcí, která pro klíč $a \in Keys(\mathcal{F}^0)$ a vstup $x \in \{0,1\}^l$ vrací hodnotu $\mathcal{F}_a^0(x) = \mathcal{F}^0(a, x) \in \{0,1\}^L$. Označíme $Rand^{l \to L}$ množinu všech funkcí $\{0,1\}^l \to \{0,1\}^L$. Nechť $\mathcal{D}$ je distinguisher, který má k dispozici orákulum $\mathcal{O}^{\mathcal{F}^b}(\cdot)$ a je omezen výpočetním časem t a počtem dotazů na orákulum q. Jeho útok na PRF definujeme jako experiment $Exp^{\text{prf}}_{\text{PRF},\mathcal{D}}$:

- na začátku experimentu je provedena náhodná volba $b \xleftarrow{R} \{0,1\}$: pokud $b = 0$, je zvolen klíč $a \xleftarrow{R} Keys(\mathcal{F}^0)$ a $\mathcal{O}^{\mathcal{F}^0}(\cdot)$ je funkce $\mathcal{F}_a^0(\cdot)$, pokud $b = 1$, je zvolena funkce $F^1 \xleftarrow{R} Rand^{l \to L}$ a $\mathcal{O}^{\mathcal{F}^1}(\cdot)$ je funkce $\mathcal{F}^1(\cdot)$;
- distinguisher $\mathcal{D}$ posílá dotazy na orákulum $\mathcal{O}^{\mathcal{F}^b}(\cdot)$ tvaru x a orákulum vrací odpovědi $y = \mathcal{F}^b(x)$;
- na konci běhu $\mathcal{D}$ odpoví hodnotu b', což je jeho odhad na bit b.

Řekneme, že distinguisher $\mathcal{D}$ je úspěšný, pokud platí $b = b'$. Advantage distinguishera $\mathcal{D}$ při útoku na PRF vyjádříme jako:

$$\mathbf{Adv}^{\text{prf}}_{\text{PRF}}(\mathcal{D}) = |\Pr[\mathcal{D} \to 1 | b = 0] - \Pr[\mathcal{D} \to 1 | b = 1]|$$

Definice 5 ((t, q, ϵ)-bezpečnost PRF) *Řekneme, že rodina pseudonáhodných funkcí PRF je (t, q, ϵ)-bezpečná, pokud neexistuje žádný distinguisher $\mathcal{D}$ omezený časem t a počtem dotazů q, jehož advantage v experimentu $Exp^{\text{prf}}_{\text{PRF},\mathcal{D}}$ je alespoň ϵ.*

Podpisové schéma SIG Podpisové schéma SIG definujeme jako trojici $(Gen, Sign, Ver)$, kde Gen je algoritmus na generování párových klíčů, který pro vstup 1^l vytvoří dvojici (pk, sk), kde pk je veřejný podpisový klíč a sk je soukromý podpisový klíč, $Sign$ je podpisový algoritmus, který má na vstupu zprávu M a klíč sk a vrací podpis $x = Sign_{sk}(M)$, a Ver je verifikační algoritmus, který pro zprávu M, kandidáta na podpis x' a klíč pk vrací bit $b = Ver_{pk}(M, x')$, kde $b = 1$, pokud Ver podpis přijme jako platný, a $b = 0$, pokud ho odmítne jako neplatný. Nechť $\mathcal{F}$ je forger, který má k dispozici podpisové orákulum $\mathcal{O}^{\text{sign}}(\cdot)$ a je omezen výpočetním časem t a počtem dotazů na orákulum q. Jeho útok na SIG definujeme jako experiment $Exp^{\text{forge}}_{\text{SIG},\mathcal{F}}$:

- na začátku experimentu je provedena náhodná volba $(pk, sk) \xleftarrow{R} Gen(1^l)$ a forger $\mathcal{F}$ dostane pk;
- forger $\mathcal{F}$ posílá dotazy na orákulum $\mathcal{O}^{\text{sign}}(\cdot)$ tvaru M_i a orákulum vrací odpovědi $x_i = Sign_{sk}(M_i)$;
- na konci běhu $\mathcal{F}$ odpoví dvojici (M, x), což je jeho pokus o padělek.

Řekneme, že forger $\mathcal{F}$ je úspěšný, pokud platí $Ver_{sk}(M, x) = 1$ a zároveň pro všechna $i : M \neq M_i$.

Definice 6 ((t, q, ϵ)-bezpečné schéma SIG) *Řekneme, že podpisové schéma SIG je (t, q, ϵ)-bezpečné, pokud neexistuje žádný forger $\mathcal{F}$ omezený časem t a počtem dotazů q, který vytvoří platný padělek s pravděpodobností alespoň ϵ.*

Schéma MAC Schéma MAC definujeme jako trojici (Gen, Mac, Ver), kde Gen je algoritmus na generování klíče k, Mac je tagovací algoritmus, který má pro klíč k a zprávu M vrací tag $\tau = Mac_k(M)$, a Ver je verifikační algoritmus, který pro klíč k zprávu M a kandidáta na tag τ vrací bit $b = Ver_k(M, \tau)$, kde $b = 1$, pokud Ver přijme tag jako platný, a $b = 0$, pokud ho odmítne jako neplatný. Nechť $\mathcal{F}$ je forger, který má k dispozici podpisové orákulum $\mathcal{O}^{\text{mac}}(\cdot)$ a je omezen výpočetním časem t a počtem dotazů na orákulum q. Jeho útok na MAC definujeme jako experiment $Exp^{\text{forge}}_{\text{MAC},\mathcal{F}}$:

- na začátku experimentu je provedena náhodná volba $k \xleftarrow{R} Gen(1^l)$;

- forger $\mathcal{F}$ posílá dotazy na orákulum $\mathcal{O}^{\text{mac}}(\cdot)$ tvaru M_i a orákulum vrací odpovědi $\tau_i = Mac_k(M_i)$;
- na konci běhu $\mathcal{F}$ odpoví dvojici (M, τ), což je jeho pokus o padělek.

Řekneme, že forger $\mathcal{F}$ je úspěšný, pokud platí $Ver_k(M, \tau) = 1$ a zároveň pro všechna $i : M \neq M_i$.

Definice 7 ((t, q, ϵ)-bezpečné schéma MAC) *Řekneme, že schéma MAC je (t, q, ϵ)-bezpečné,pokud neexistuje žádný forger $\mathcal{F}$ omezený časem t a počtem dotazů q, který vytvoří platný padělek s pravděpodobností alespoň ϵ.*

5.3 Pojmy použité v důkazu bezpečnosti Σ_0

Následující dvě definice budou použity v důkazu (t, q, δ, ϵ)-SK-bezpečnosti Σ_0. Jedná se o zásadní pojmy, na kterých celý důkaz stojí.

Definice 8 (Algoritmy s ϵ-blízkým rozdělením výstupů) *Nechť $\mathcal{D}$ a $\mathcal{D}'$ jsou dva pravděpodobnostní algoritmy s výstupem 0 nebo 1. Řekneme, že jsou to algoritmy s ϵ-blízkým rozdělením výstupů (budeme značit $\mathcal{D} \approx_\epsilon \mathcal{D}'$) právě tehdy, když.*

$$|\Pr[\mathcal{D} \to 1] - \Pr[\mathcal{D}' \to 1]| < \epsilon$$

Definice 9 (Událost GUESS) *Nechť $\hat{\mathcal{S}}$ je jednou z verzí simulátorů (jak jsou definované v sekci 4.3) a nechť $\mathcal{A}$ je útočník na Σ_0. Řekneme, že v běhu $\hat{\mathcal{S}}(\mathcal{A})$ nastala událost GUESS, jestliže jsou splněny následující podmínky:*

1. *Útočník $\mathcal{A}$ v tomto běhu iniciuje alespoň T sezení (T-té sezení označíme (I_0, s_0)), kde T je parametr zvolený simulátorem $\hat{\mathcal{S}}$ v úvodu simulace.*
2. *Pokud R_0 je identita náhodného účastníka zvolená simulátorem $\hat{\mathcal{S}}$ v úvodu simulace, pak buď*
   (a) *$\mathcal{A}$ zvolí jako své testovací sezení (I_0, s_0) a toto sezení se dokončí s peerem R_0, nebo*
   (b) *$\mathcal{A}$ zvolí jako své testovací sezení (R_0, s_0) a toto sezení se dokončí s peerem I_0*

Poznámka Událost GUESS je tedy případ, kdy simulátor správně uhodl, které sezení bude testovací a kdo v něm bude v roli respondéra.

Kapitola 6

Důkaz bezpečnosti Σ_0 v kontextu konkrétní bezpečnosti

Náplní této kapitoly bude podrobný důkaz konkrétní bezpečnosti protokolu Σ_0 za podmínek, že použitá grupa splňuje předpoklad DDH a že protokol používá bezpečné primitivy popsané v předešlé kapitole. Toto vyjádříme formou věty:

Věta 1 *Pokud protokol výměny klíče Σ_0 používá grupu splňující $(t_{\mathrm{DDH}}, \epsilon_{\mathrm{DDH}})$-DDH-předpoklad, $(t_{\mathrm{PRF}}, q_{\mathrm{PRF}}, \epsilon_{\mathrm{PRF}})$-bezpečnou rodinu pseudonáhodných funkcí PRF a $(t_{\mathrm{SIG}}, q_{\mathrm{SIG}}, \epsilon_{\mathrm{SIG}})$-bezpečné podpisové schéma SIG, pak je tento protokol (t, q, δ, ϵ)-SK-bezpečný, kde $t = max\{t_{\mathrm{SIG}}, t_{\mathrm{DDH}}, t_{\mathrm{PRF}}\} - \mathcal{O}(1)$, q je soubor $(q_{\mathrm{state}}, q_{\mathrm{sk}}, q_{\mathrm{corr}}, q_{\mathrm{resp}})$, $\epsilon = \epsilon_{\mathrm{PRF}} \cdot (\frac{1}{2} + 2mn) + \epsilon_{\mathrm{DDH}} \cdot 2mn$ pro m apriorní horní mez počtu sezení iniciovaných útočníkem a n počet účastníků protokolu a $\delta = 1 - \epsilon_{\mathrm{SIG}}$. Navíc $q_{\mathrm{state}} = m - 2 = q_{\mathrm{sk}}$, $q_{\mathrm{resp}} = q_{\mathrm{SIG}} - 1$ a $q_{\mathrm{corr}} = n - 2$.*

Postup důkazu jsme již uvedli v části 4.2. Připomeňme, že budeme dokazovat bod 1. a bod 2. z definice 4, které budeme nazývat *vlastnost 1* a *vlastnost 2*. V důkazu vlastnosti 2 budou stěžejní simulátory popsané v části 4.3.

Značení V následujících důkazech bude $(\ldots)_{\mathrm{ID}\rightarrow}$ značit zprávu odeslanou účastníkem ID a $(\ldots)_{\rightarrow\mathrm{ID}}$ zprávu přijatou účastníkem ID.

Poznámka V uvedené větě je záměrně vynechán předpoklad bezpečnosti schématu MAC. Je to z toho důvodu, že pro zjednodušení explicitního vyjádření bezpečnosti předpokládáme, že protokol místo schématu MAC používá rodinu pseudonáhodných funkcí PRF (protože každá bezpečná pseudonáhodná funkce je zároveň bezpečný MAC). Pro každou roli PRF je samozřejmě použit jiný klíč. Zároveň v praxi se v mnoha případech PRF takto opravdu používá.

6.1 Důkaz vlastnosti 1

Nejprve si vlastnost 1 z definice (t, q, δ, ϵ)-SK-bezpečnosti připomeňme. Ta nám říká, že pokud dva nezkorumpovaní účastníci, které si zde označíme ID_i a ID_r, dokončí shodná sezení $(\mathrm{ID}_i, s_0, \mathrm{ID}_r)$ a $(\mathrm{ID}_r, s_0, \mathrm{ID}_i)$ v běhu protokolu Σ_0 při útoku útočníka $\mathcal{A}$ omezeného výpočetním časem t a počty dotazů q, pak pravděpodobnost, že výstup session key byl pro obě sezení stejný, je větší než δ. My tvrdíme, že $\delta = 1 - \epsilon_{\mathrm{SIG}}$, $t = t_{\mathrm{SIG}} - \mathcal{O}(1)$ a $q_{\mathrm{resp}} = q_{\mathrm{SIG}} - 1$.

Důkaz: Nejprve je třeba si uvědomit, že session key je deterministicky odvozen ze sdíleného DH tajemství. Stačí tedy ukázat, že oba účastníci odvodí stejné DH tajemství s pravděpodobností větší než $1 - \epsilon_{\mathrm{SIG}}$.

Označme si tedy DH hodnotu odeslanou v úvodní zprávě účastníkem ID_i jako u_i a DH hodnotu přijatou v úvodní zprávě účastníkem ID_r jako u_r. Podobně označme DH hodnotu odeslanou v odpovědní zprávě účastníkem ID_r jako v_r a DH hodnotu přijatou v odpovědní zprávě účastníkem ID_i jako v_i. Aby oba účastníci vypočítali stejné DH tajemství, musí platit $(u_i, v_i) = (u_r, v_r)$.

Účastníci dokončili sezení, tedy oba ověřili přijaté podpisy. Protože k vytvoření podpisu je použit unikátní sessionID s, útočník nemohl podstrčit nějaký starší platný podpis a pokud tedy $(u_i, v_i) \neq (u_r, v_r)$, musel jeden z těchto podpisů padělat. BÚNO budeme předpokládat, že to byl podpis respondéra ID_r. Ten vytvořil podpis $\mathrm{SIG}_r(\text{`1'}, s, u_r, v_r)$, zatímco iniciátor $\mathrm{ID}_i$ ověřil podpis $\mathrm{SIG}_r(\text{`1'}, s, u_i, v_i)$.

Ukážeme, že pokud by útočník dokázal vytvořit platný $\mathrm{SIG}_r(\text{`1'}, s, u_i, v_i)$, můžeme ho použít k sestrojení úspěšného forgeru $\mathcal{F}_{\mathrm{SIG}}$. Ten má k dispozici orákulum $\mathcal{O}^{sign}(\cdot)$ a využije útočníka $\mathcal{A}$ následovně:

- $\mathcal{O}^{sign}(\cdot)$ zvolí dvojici klíčů (pk, sk), pošle pk forgeru $\mathcal{F}_{\mathrm{SIG}}$ a ten spustí útočníka $\mathcal{A}$;
- když $\mathcal{A}$ aktivuje ID_i jako iniciátora, zvolí $\mathcal{F}_{\mathrm{SIG}}$ hodnoty s a $x_i \xleftarrow{R} \mathbb{Z}_q$, položí $u_i = g^{x_i}$ a pošle útočníkovi $(s, u_i)_{\mathrm{ID}_i \rightarrow}$;
- $\mathcal{A}$ zvolí u_r a pošle $(s, u_r)_{\rightarrow \mathrm{ID}_r}$;
- $\mathcal{F}_{\mathrm{SIG}}$ zvolí $x_r \xleftarrow{R} \mathbb{Z}_q$, položí $v_r = g^{x_r}$, pošle dotaz na $\mathcal{O}^{sign}(\cdot)$ tvaru $M = (\text{`1'}, s, u_r, v_r)$, dostane odpověď $\mathrm{SIG}_{sk}(\text{`1'}, s, u_r, v_r)$ a pošle útočníkovi $(s, v_r, \mathrm{SIG}_{sk}(\text{`1'}, s, u_r, v_r), \ldots)_{\mathrm{ID}_r \rightarrow}$;
- $\mathcal{A}$ se nyní snaží vytvořit platný podpis $\mathrm{SIG}_{sk}(\text{`1'}, s, u_i, v_i)$ a posílá dotazy na své orákulum $\mathcal{O}^{resp}(\cdot)$ tvaru (s_j, u_j) pro jím volené hodnoty s_j a u_j, kde $j \leq q_{\mathrm{resp}}$;

- $\mathcal{F}_{\text{SIG}}$ pro každý dotaz (s_j, u_j) zvolí v_j, pošle dotaz na $\mathcal{O}^{sign}(\cdot)$ tvaru $M_j = (\text{`1'}, s_j, u_j, v_j)$, dostane odpověď $\text{SIG}_{sk}(\text{`1'}, s_j, u_j, v_j)$ a útočníkovi pošle $(s_j, v_j, \text{SIG}_{sk}(\text{`1'}, s_j, u_j, v_j), \ldots)_{\text{ID}_r \rightarrow}$;
- $\mathcal{A}$ vytvoří $\text{SIG}_{sk}(\text{`1'}, s, u_i, v_i)$ a pošle $(s, v_i, \text{SIG}_{sk}(\text{`1'}, s, u_i, v_i), \ldots)_{\rightarrow \text{ID}_i}$;
- $\mathcal{F}_{\text{SIG}}$ odpoví dvojicí $(\text{`1'}, s, u_i, v_i), \text{SIG}_{sk}(\text{`1'}, s, u_i, v_i)$.

Vidíme, že $t = t_{\text{SIG}} - \mathcal{O}(1)$ a $q_{\text{resp}} = q_{\text{SIG}} - 1$. Dále platí, že pravděpodobnost, že ID_i a ID_r vypočtou různá DH tajemství, je rovna pravděpodobnosti, že námi sestrojený forger $\mathcal{F}_{\text{SIG}}$ uspěje při útoku na SIG. Protože ale předpokládáme, že SIG je $(t_{\text{SIG}}, q_{\text{SIG}}, \epsilon_{\text{SIG}})$-bezpečné, musí být tato pravděpodobnost menší než ϵ_{SIG}.

■

Poznámka Vlastnost 1 vyjadřuje s jakou (jak nízkou) pravděpodobností mohl útočník nepozorovaně narušit výměnu tajného klíče. Každý z účastníků by v té chvíli věřil, že sdílí klíč s tím druhým, ale ve skutečnosti by každý dohodl jiný klíč s útočníkem, který by tak získal přehled o veškeré komunikaci.

6.2 Důkaz vlastnosti 2

Jak jsme již uvedli dříve, důkaz vlastnosti 2 je značně složitější. Jednotlivé argumenty vyjádříme formou lemmat, která dokážeme zvlášť. V lemmatech 1 a 5 ukážeme několik vlastností souvisejících s prvky pro zajištění ochrany autentičnosti a integrity, ostatní lemmata se budou týkat jednotlivých simulací a vztahy mezi nimi.

Opět si nejprve připomeneme vlastnost 2 z definice (t, q, δ, ϵ)-SK-bezpečnosti. Ta říká, že pro advantage útočníka $\mathcal{A}$ s omezením výpočetních zdrojů t a q útočícího na protokol Σ_0 platí $\mathbf{Adv}^{\text{test}}_{\Sigma_0}(\mathcal{A}) < \epsilon$. My tvrdíme, že $\epsilon = \epsilon_{\text{PRF}} \cdot (\frac{1}{2} + 2mn) + \epsilon_{\text{DDH}} \cdot 2mn$, $t = max\{t_{\text{DDH}}, t_{\text{PRF}}\} - \mathcal{O}(1)$.

Advantage útočníka $\mathcal{A}$ Ještě než uvedeme jednotlivá lemmata, vyjádříme si advantage $\mathcal{A}$ pomocí hodnot $\text{Pr}_{\text{REAL}}(\mathcal{A})$ (pravděpodobnost, že $\mathcal{A}$ neuhádl, že challenge session key byl skutečný klíč) a $\text{Pr}_{\text{RAND}}(\mathcal{A})$ (pravděpodobnost, že $\mathcal{A}$ uhádl, že challenge session key byla náhodná hodnota), které definujeme následovně:

$$\text{Pr}_{\text{REAL}}(\mathcal{A}) = \text{Pr}[\mathcal{A} \rightarrow 1 | b = 0], \text{Pr}_{\text{RAND}}(\mathcal{A}) = \text{Pr}[\mathcal{A} \rightarrow 1 | b = 1]$$

Použijeme-li tyto hodnoty, dostaneme, že:

$$\mathbf{Adv}^{\text{test}}_{\Sigma_0}(\mathcal{A}) = |\text{Pr}_{\text{REAL}}(\mathcal{A}) - \text{Pr}_{\text{RAND}}(\mathcal{A})| < \epsilon$$

Lemma 1 *Za předpokladu $(t_{\mathrm{SIG}}, q_{\mathrm{SIG}}, \epsilon_{\mathrm{SIG}})$-bezpečnosti schématu SIG pro každého útočníka $\mathcal{A}$ na Σ_0 omezeného výpočetním časem $t = t_{\mathrm{SIG}} - \mathcal{O}(1)$ a počtem dotazů $q_{\mathrm{resp}} = q_{\mathrm{SIG}} - 1$ platí:*

1. *Předpokládáme běžný běh útočníka $\mathcal{A}$, ve kterém zvolí testovací sezení s veřejným výstupem (P, s, Q), kde P je v roli iniciátora. Pak:*
   (a) *P a Q nejsou nikdy zkorumpováni před vypršením testovacího sezení.*
   (b) *Útočník $\mathcal{A}$ na sezení (P, s) a (Q, s) nikdy nepoužije útok session state reveal.*
   (c) *S pravděpodobností větší než $1 - \epsilon_{\mathrm{SIG}}$ je sezení (Q, s) aktivováno úvodní zprávou poslanou sezením (P, s) a Q je v roli respondéra.*
   (d) *S pravděpodobností větší než $1 - \epsilon_{\mathrm{SIG}}$ sezení (P, s) obdrží odpovědní zprávu poté, co bylo (Q, s) pro Q v roli respondéra, a tato zpráva bude obsahovat stejnou DH hodnotu jako odpovědní zpráva poslaná sezením (Q, s).*
   (e) *Sezení (P, s) se nikdy nepřeruší.*
2. *Zcela stejně vše platí, pokud $\mathcal{A}$ zvolí testovací sezení s veřejným výstupem (Q, s, P), kde Q je v roli respondéra.*

Důkaz: Vlastnosti (a), (b) a (e) plynou přímo z definice testovacího sezení. Zbývá tedy ukázat vlastnosti (c) a (d).

Protože testovací sezení bylo dokončeno a jeho veřejný výstup je tvaru (P, s, Q), musel účastník P obdržet odpovědní zprávu obsahující (mimo jiné) sessionID s, DH hodnotu v_i, identitu Q a podpis $\mathrm{SIG}_Q(\text{`1'}, s, g^x, v_i)$, kde g^x je DH hodnota účastníka P, a tento podpis následně ověřit veřejným klíčem účastníka Q. Tedy podpis musel být buď skutečně vytvořen účastníkem Q a tedy muselo dojít pomocí úvodní zprávy (s, g^x) odeslané sezením (P, s) k aktivování sezení (Q, s) pro Q v roli respondéra, nebo musel útočník podpis padělat. Jak jsme ukázali v důkazu vlastnosti 1, v tom případě by bylo možné na základě útočníka $\mathcal{A}$ sestrojit forger útočící na schéma SIG. Tedy pravděpodobnost, že k padělání dojde a (Q, s) nebylo aktivováno úvodní zprávou (s, g^x), musí být menší než ϵ_{SIG} a $t = t_{\mathrm{SIG}} - \mathcal{O}(1)$.

Pokud tedy (Q, s) bylo aktivováno pro Q v roli respondéra, z vlastnosti 1 opět dostáváme, že s pravděpodobností větší než ϵ_{SIG} byla v odpovědní zprávě přijaté sezením (P, s) stejná hodnota jako ve zprávě odeslané sezením (Q, s).

■

Poznámka Lemma 1 uvádíme především jako ucelený pohled na vlastnosti testovacího sezení plynoucí z použití bezpečného podpisového schématu SIG. Vlastnosti plynoucí přímo z definice jsou uvedeny právě pro ucelenost. Můžeme jeho znění interpretovat tak, že útočník nemohl předem do testovacího sezení „podstrčit“ svou DH hodnotu a nemohl zajistit, že sezení bylo dokončeno s jiným peerem, než účastník věří (až na malou pravděpodobnost, omezenou schématem SIG).

Lemma 2 *Za* $(t_{\mathrm{DDH}}, \epsilon_{\mathrm{DDH}})$*-DDH-předpokladu pro použitou grupu pro každého útočníka* $\mathcal{A}$ *útočícího na* Σ_0 *a omezeného výpočetním časem* $t = t_{\mathrm{DDH}} - \mathcal{O}(1)$ *platí:*

$$\hat{\mathcal{S}}{-}\mathrm{RAND}(\mathcal{A}) \approx_{\epsilon_{\mathrm{DDH}}} \hat{\mathcal{S}}{-}\mathrm{HYBR}(\mathcal{A})$$

Důkaz: Budeme dokazovat, že pro každého útočníka $\mathcal{A}$ platí nerovnost

$$\left|\Pr[\hat{\mathcal{S}}{-}\mathrm{RAND}(\mathcal{A}) \to 1] - \Pr[\hat{\mathcal{S}}{-}\mathrm{HYBR}(\mathcal{A}) \to 1]\right| < \epsilon_{\mathrm{DDH}}$$

Cílem tedy bude ukázat, že pokud by existoval útočník, pro kterého by byl rozdíl daných pravděpodobností roven alespoň ϵ_{DDH}, bylo by s jeho použitím možné sestrojit distinguisher $\mathcal{D}_{\mathrm{DDH}}$ proti $(t_{\mathrm{DDH}}, \epsilon_{\mathrm{DDH}})$-DDH-předpokladu, který by pro dané g^x, g^y dokázal rozlišovat g^{xy} od g^r pro r náhodné s advantage alespoň ϵ_{DDH}.

Úkolem distinguishera je tedy pro vstupní trojici (g^x, g^y, z) rozhodnout, zda $z = g^{xy}$, nebo $z = g^r$ pro r náhodné. Svůj útok provede tak, že bude simulovat $\hat{\mathcal{S}}{-}\mathrm{RAND}$ pro útočníka $\mathcal{A}$ s tím rozdílem, že pro sezení (I_0, s_0) a (R_0, s_0) použije jako DH hodnoty jednotlivých účastníků vstupní hodnoty g^x a g^y a klíče k_1 a k_0 definuje s použitím vstupní hodnoty z jako $k_1 \leftarrow \mathrm{PRF}_z(1)$ a $k_0 \leftarrow r()$. Ostatní akce pak vykonává podle pravidel $\hat{\mathcal{S}}{-}\mathrm{RAND}$. Na konci běhu pak odpoví stejnou hodnotou, jako odpověděl $\mathcal{A}$. V případě události přerušení zastaví simulaci. Odsud plyne, že $t = t_{\mathrm{DDH}} - \mathcal{O}(1)$.

V případě, že vstupní hodnota $z = g^{xy}$, útočníkovi $\mathcal{A}$ se simulace $\mathcal{D}_{\mathrm{DDH}}(\mathcal{A})$ jeví stejně jako $\hat{\mathcal{S}}{-}\mathrm{RAND}(\mathcal{A})$. Jediná změna nastala v T-tém sezení, kde nebyly voleny náhodné hodnoty jako DH exponenty pro výpočet DH hodnot, ale jako DH hodnoty byly použity g^x a g^y a následkem toho se DH exponenty neobjevily ve vnitřním stavu sezení. Hodnoty g^x a g^y ovšem mají stejné rozdělení, jako DH hodnoty v běžném běhu protokolu, a pokud by útočník použil session state reveal, došlo by k přerušení. Tedy chování útočníka není ničím ovlivněno a dostáváme

$$\Pr[\hat{\mathcal{S}}{-}\mathrm{RAND}(\mathcal{A}) \to 1] = \Pr[\mathcal{D}_{\mathrm{DDH}} \to 1 | z = g^{xy}] = \Pr[\mathcal{D}^{Q_1} \to 1]$$

V případě, že $z = g^r$ pro r náhodné se pak útočníkovi simulace jeví stejně jako $\hat{\mathcal{S}}-\mathrm{HYBR}(\mathcal{A})$. A s použitím stejných argumentů jako výše dostáváme

$$\Pr[\hat{\mathcal{S}}-\mathrm{HYBR}(\mathcal{A}) \to 1] = \Pr[\mathcal{D}_{\mathrm{DDH}} \to 1 | z = g^r, r \text{ náhodné}] = \Pr[\mathcal{D}^{Q_0} \to 1]$$

Z $(t_{\mathrm{DDH}}, \epsilon_{\mathrm{DDH}})$-DDH-předpokladu tedy máme, že pro každého útočníka $\mathcal{A}$ platí:

$$\left|\Pr[\hat{\mathcal{S}}-\mathrm{RAND}(\mathcal{A}) \to 1] - \Pr[\hat{\mathcal{S}}-\mathrm{HYBR}(\mathcal{A}) \to 1]\right| < \epsilon_{\mathrm{DDH}}$$

■

Lemma 3 *Pro libovolného útočníka $\mathcal{A}$ útočícího na Σ_0 je pravděpodobnost události GUESS při běhu simulátoru $\hat{\mathcal{S}}-\mathrm{RAND}(\mathcal{A})$ alespoň $\frac{1}{m \cdot n}$, kde m je apriorní horní mez počtu sezení iniciovaných útočníkem $\mathcal{A}$ a n je počet účastníků protokolu.*

Důkaz: Nejprve si ukážeme, jaká je pravděpodobnost události GUESS při běhu obecného simulátoru $\mathcal{S}(\mathcal{A})$. Protože v tomto případě útočník vždy zvolí testovací sezení, pak pokud zvolíme náhodné sezení (I_0, s_0) a náhodného účastníka R_0, je pravděpodobnost, že veřejný výstup testovacího sezení zvoleného útočníkem $\mathcal{A}$ v běhu $\mathcal{S}(\mathcal{A})$ bude $(\text{'0'}, I_0, s_0, R_0)$ nebo $(\text{'0'}, I_0, s_0, R_0)$, kde příznaky 0 a 1 značíme, zda se jedná o veřejný výstup iniciátora nebo respondéra, rovna alespoň $\frac{1}{m \cdot n}$.

Nyní si označíme jako $\hat{\mathcal{S}}-\mathrm{RAND}'$ simulátor, který se chová přesně jako $\hat{\mathcal{S}}-\mathrm{RAND}$, ale nezastaví svůj běh v případě události přerušení. Platí, že za běhu $\hat{\mathcal{S}}-\mathrm{RAND}'(\mathcal{A})$ je jako challenge session key použita náhodná hodnota, zatímco za běhu $\mathcal{S}(\mathcal{A})$ je to skutečný session key. To ale neovlivní, jakým způsobem $\mathcal{A}$ volí testovací sezení, tedy pravděpodobnost události GUESS při běhu $\hat{\mathcal{S}}-\mathrm{RAND}'(\mathcal{A})$ je stejná jako při běhu $\mathcal{S}(\mathcal{A})$, tedy alespoň $\frac{1}{m \cdot n}$. Uvažujme nyní pevnou množinu voleb, které $\hat{\mathcal{S}}-\mathrm{RAND}'$ při simulaci protokolu pro útočníka $\mathcal{A}$ provede a které přivedou $\hat{\mathcal{S}}-\mathrm{RAND}'(\mathcal{A})$ k události GUESS. Pokud se podíváme na obyčejný běh $\hat{\mathcal{S}}-\mathrm{RAND}$ se stejnou množinou voleb, také dojde k události GUESS, protože dle lemma 1 nedojde k události přerušení. Z toho vyplývá, že při použití této množiny voleb se běh simulátoru $\hat{\mathcal{S}}-\mathrm{RAND}(\mathcal{A})$ neliší od běhu $\hat{\mathcal{S}}-\mathrm{RAND}'(\mathcal{A})$. Tedy každá množina voleb, která k události GUESS přivede $\hat{\mathcal{S}}-\mathrm{RAND}'(\mathcal{A})$, k ní přivede také $\hat{\mathcal{S}}-\mathrm{RAND}(\mathcal{A})$. Odsud dostáváme, že

$$\Pr[\text{GUESS při } \hat{\mathcal{S}}-\mathrm{RAND}(\mathcal{A})] \geq \Pr[\text{GUESS při } \hat{\mathcal{S}}-\mathrm{RAND}'(\mathcal{A})] \geq \frac{1}{m \cdot n}$$

■

Lemma 4 *Za* $(t_{\mathrm{DDH}}, \epsilon_{\mathrm{DDH}})$*-DDH-předpokladu pro použitou grupu pro každého útočníka* $\mathcal{A}$ *útočícího na* Σ_0 *a omezeného výpočetním časem* $t = t_{\mathrm{DDH}} - \mathcal{O}(1)$ *platí:*

$$\left|\Pr[\text{GUESS } \textit{při } \hat{\mathcal{S}}{-}\text{RAND}(\mathcal{A})] - \Pr[\text{GUESS } \textit{při } \hat{\mathcal{S}}{-}\text{HYBR}(\mathcal{A})]\right| < \epsilon_{\mathrm{DDH}}$$

Důkaz: Ukážeme, že pokud by byl rozdíl uvedených pravděpodobností alespoň ϵ_{DDH}, mohli bychom sestrojit distinguisher ${\mathcal{D}'}_{\mathrm{DDH}}$ proti $(t_{\mathrm{DDH}}, \epsilon_{\mathrm{DDH}})$-DDH-předpokladu, jehož advantage by byla rovna alespoň ϵ_{DDH}.

Tento distinguisher ${\mathcal{D}'}_{\mathrm{DDH}}$ by ke svému útoku použil distinguisher $\mathcal{D}_{\mathrm{DDH}}$ popsaný v důkazu lemma 2 tak, že by po obdržení vstupní trojice (g^x, g^y, z) spustil $\mathcal{D}_{\mathrm{DDH}}$, ale na konci běhu by odpověděl bit 1 právě tehdy, když by nastala v běhu $\mathcal{D}_{\mathrm{DDH}}$ událost GUESS (připomeňme, že $\mathcal{D}_{\mathrm{DDH}}$ simuluje útočníkovi $\mathcal{A}$ protokol stejně jako simulátor $\hat{\mathcal{S}}{-}\text{RAND}$, pouze pro T-té sezení nevolí účastníkům DH hodnoty, ale použije g^x a g^y, a pro výpočet klíčů k_1 a k_0 použije z). Stejně jako pro lemma 2 dostáváme, že $t = t_{\mathrm{DDH}} - \mathcal{O}(1)$.

V případě, že $z = g^{xy}$, provádí $\mathcal{D}_{\mathrm{DDH}}$ simulaci $\hat{\mathcal{S}}{-}\text{RAND}$ (stejně jako v lemmatu 2), tedy ${\mathcal{D}'}_{\mathrm{DDH}}$ odpoví bit 1 se stejnou pravděpodobností, jakou má událost GUESS při běhu $\hat{\mathcal{S}}{-}\text{RAND}$. Dostáváme tedy

$$\Pr[\text{GUESS při } \hat{\mathcal{S}}{-}\text{RAND}(\mathcal{A})] = \Pr[{\mathcal{D}'}_{\mathrm{DDH}} \to 1 | z = g^{xy}] = \Pr[{\mathcal{D}'}^{Q_1} \to 1]$$

V případě, že $z = g^r$ pro r náhodné, provádí $\mathcal{D}_{\mathrm{DDH}}$ simulaci $\hat{\mathcal{S}}{-}\text{HYBR}$, tedy ${\mathcal{D}'}_{\mathrm{DDH}}$ odpoví bit 1 se stejnou pravděpodobností, jakou má GUESS při běhu $\hat{\mathcal{S}}{-}\text{HYBR}$, a dostáváme

$$\Pr[\text{GUESS při } \hat{\mathcal{S}}{-}\text{HYBR}(\mathcal{A})] = \Pr[{\mathcal{D}'}_{\mathrm{DDH}} \to 1 | z = g^{r}] = \Pr[{\mathcal{D}'}^{Q_0} \to 1]$$

S použitím $(t_{\mathrm{DDH}}, \epsilon_{\mathrm{DDH}})$-DDH-předpokladu tedy dostaneme

$$\left|\Pr[\text{GUESS při } \hat{\mathcal{S}}{-}\text{RAND}(\mathcal{A})] - = \Pr[\text{GUESS při } \hat{\mathcal{S}}{-}\text{HYBR}(\mathcal{A})]\right| < \epsilon_{\mathrm{DDH}}$$

■

Lemma 5 *Za předpokladu* $(t_{\mathrm{PRF}}, q_{\mathrm{PRF}}, \epsilon_{\mathrm{PRF}})$*-bezpečnosti rodiny PRF pro každého útočníka* $\mathcal{A}$ *útočícího na* Σ_0 *a omezeného časem* $t = t_{\mathrm{PRF}} - \mathcal{O}(1)$ *platí, že pokud nastala událost GUESS při běhu* $\hat{\mathcal{S}}{-}\text{HYBR}(\mathcal{A})$*, pak s pravděpodobností větší než* $1 - \epsilon_{\mathrm{PRF}}$ *platí:*

1. *Pokud bylo sezení* (I_0, s_0) *zvoleno útočníkem* $\mathcal{A}$ *jako testovací, pak sezení* (R_0, s_0) *je s ním shodné.*
2. *Pokud bylo sezení* (R_0, s_0) *zvoleno útočníkem* $\mathcal{A}$ *jako testovací, pak sezení* (I_0, s_0) *je s ním shodné.*

Důkaz: Dokážeme pouze bod 1., bod 2. se dokazuje zcela analogicky. Předpokládáme, že při běhu $\hat{\mathcal{S}}-\mathrm{HYBR}(\mathcal{A})$ nastala událost GUESS a že bylo zvoleno testovací sezení (I_0, s_0). Z definice GUESS máme, že jeho peerem je R_0. Pro sezení (R_0, s_0) platí, že pokud není dokončené, je shodné se sezením (I_0, s_0).

V případě, že (R_0, s_0) je dokončené s veřejným výstupem (R_0, s_0, ID), je shodné s (I_0, s_0) právě tehdy, když $\mathrm{ID} = I_0$. Aby se (R_0, s_0) mohlo dokončit, musel účastník R_0 ověřit hodnotu $\mathrm{MAC}_{k_1}(\text{‘0‘}, s_0, \mathrm{ID})$ pomocí klíče $k_1 = \mathrm{PRF}_k(1)$, kde k je náhodný klíč zvolený simulátorem $\hat{\mathcal{S}}-\mathrm{HYBR}$, který není útočníkovi $\mathcal{A}$ nikdy poskytnut. V okamžiku ověření tohoto MACu existovaly v protokolu pouze dvě instance MAC_{k_1} (žádné jiné nemohly obsahovat sessionID s). Byly to $\mathrm{MAC}_{k_1}(\text{‘1‘}, s_0, R_0)$ a $\mathrm{MAC}_{k_1}(\text{‘0‘}, s_0, I_0)$. Tedy buď $\mathrm{ID} = I_0$, nebo $\mathrm{ID} \neq I_0$ a znamená to, že útočníkovi $\mathcal{A}$ se podařilo $\mathrm{MAC}_{k_1}(\text{‘0‘}, s_0, \mathrm{ID})$ padělat. Pokud útočník $\mathrm{MAC}_{k_1}(\text{‘0‘}, s_0, \mathrm{ID})$ skutečně padělal, můžeme ho využít k forgeru proti schématu MAC_{k_1} s klíčem $k_1 = \mathrm{PRF}_k(1)$ a ten dále využít k sestrojení distinguisheru proti schématu PRF nabo k sestrojení forgeru proti MAC s libovolným klíčem.

Forger $\mathcal{F}_{\mathrm{MAC}_{k_1}}$ proti MAC_{k_1} bude pracovat na základě $\hat{\mathcal{S}}-\mathrm{HYBR}$ a bude mít k dispozici orákulum $\mathcal{O}_{k_1}^{\mathrm{mac}}(\cdot)$. Bude postupovat následovně:

- $\mathcal{F}_{\mathrm{MAC}_{k_1}}$ zvolí hodnoty x, y, k_0, T, R_0 stejným způsobem, jako by to udělal simulátor $\hat{\mathcal{S}}-\mathrm{HYBR}$, ale na rozdíl od něj nezvolí hodnotu k_1, spustí útočníka $\mathcal{A}$ a na jeho akce reaguje stejně, jako by to dělal $\hat{\mathcal{S}}-\mathrm{HYBR}$;
- $\mathcal{A}$ iniciuje T-té sezení (I_0, s_0) s respondérem R_0;
- $\mathcal{F}_{\mathrm{MAC}_{k_1}}$ pošle úvodní zprávu $(s_0, g^x)_{I_0\rightarrow}$ jménem I_0 a $\mathcal{A}$ ji přepošle pro R_0;
- $\mathcal{F}_{\mathrm{MAC}_{k_1}}$ pošle dotaz na $\mathcal{O}_{k_1}^{\mathrm{mac}}(\cdot)$ tvaru $(\text{‘1‘}, s_0, R_0)$ a dostane odpověď $\mathrm{MAC}_{k_1}(\text{‘1‘}, s_0, R_0)$;
- $\mathcal{F}_{\mathrm{MAC}_{k_1}}$ pošle odpovědní zprávu $(s_0, g^y, R_0, \mathrm{SIG}_{R_0}(), \mathrm{MAC}_{k_1}(\text{‘1‘}, s_0, R_0))_{R_0\rightarrow}$ jménem R_0 a $\mathcal{A}$ ji přepošle pro I_0;
- $\mathcal{F}_{\mathrm{MAC}_{k_1}}$ pošle dotaz na $\mathcal{O}_{k_1}^{\mathrm{mac}}(\cdot)$ tvaru $(\text{‘0‘}, s_0, I_0)$ a dostane odpověď $\mathrm{MAC}_{k_1}(\text{‘0‘}, s_0, I_0)$;
- $\mathcal{F}_{\mathrm{MAC}_{k_1}}$ pošle závěrečnou zprávu $(s_0, I_0, \mathrm{SIG}_{I_0}(), \mathrm{MAC}_{k_1}(\text{‘0‘}, s_0, R_0))_{I_0\rightarrow}$ jménem I_0, $\mathcal{A}$ vytvoří padělek $\mathrm{MAC}_{k_1}(\text{‘0‘}, s_0, \mathrm{ID})$ a pošle pro R_0 zprávu $(s_0, \mathrm{ID}, \mathrm{SIG}_{\mathrm{ID}}, \mathrm{MAC}_{k_1}(\text{‘0‘}, s_0, \mathrm{ID}))_{\rightarrow R_0}$;
- $\mathcal{F}_{\mathrm{MAC}_{k_1}}$ na konci běhu odpoví (M, τ), kde $M = (\text{‘0‘}, s_0, \mathrm{ID})$ a $\tau = \mathrm{MAC}_{k_1}(\text{‘0‘}, s_0, \mathrm{ID})$.

Vidíme, že $\mathcal{F}_{\mathrm{MAC}_{k_1}}$ běží v čase $t + \mathcal{O}(1)$. Existují dvě možnosti, jak útočník vytváří padělky. Buď využívá znalost $k_1 = \mathrm{PRF}_k(1)$, kde k je náhodný, nebo tuto znalost nevyužívá a dokáže padělat MAC pro libovolný klíč.

Pokud platí první možnost, můžeme na základě $\mathcal{F}_{\mathrm{MAC}_{k_1}}$ sestrojit distinguisher $\mathcal{D}_{\mathrm{PRF}}$, který bude mít k dispozici orákulum $\mathcal{O}^{\mathcal{F}^b}(\cdot)$ a bude pracovat následovně:

- náhodně se zvolí bit $b \xleftarrow{R} \{0, 1\}$;
- $\mathcal{D}_{\mathrm{PRF}}$ pošle orákulu dotaz $x = 1$, položí $k_1 = \mathcal{F}^b(1)$ a spustí $\mathcal{F}_{\mathrm{MAC}_{k_1}}$;
- $\mathcal{F}_{\mathrm{MAC}_{k_1}}$ pošle dotaz na své orákulum tvaru $(\text{`1'}, s_0, R_0)$, $\mathcal{D}_{\mathrm{PRF}}$ spočte hodnotu $\mathrm{MAC}_{k_1}(\text{`1'}, s_0, R_0)$ pomocí klíče k_1 a výsledek pošle $\mathcal{F}_{\mathrm{MAC}_{k_1}}$;
- $\mathcal{F}_{\mathrm{MAC}_{k_1}}$ pošle dotaz na své orákulum tvaru $(\text{`0'}, s_0, I_0)$, $\mathcal{D}_{\mathrm{PRF}}$ spočte $\mathrm{MAC}_{k_1}(\text{`0'}, s_0, I_0)$ pomocí klíče k_1 a výsledek pošle $\mathcal{F}_{\mathrm{MAC}_{k_1}}$;
- pošle dotaz na své orákulum tvaru $(\text{`1'}, s_0, R_0)$, $\mathcal{D}_{\mathrm{PRF}}$ spočte hodnotu $\mathrm{MAC}_{k_1}(\text{`1'}, s_0, R_0)$ pomocí klíče k_1 a výsledek pošle $\mathcal{F}_{\mathrm{MAC}_{k_1}}$;
- $\mathcal{F}_{\mathrm{MAC}_{k_1}}$ odpoví (M, τ) a $\mathcal{D}_{\mathrm{PRF}}$ tuto dvojici ověří pomocí klíče k_1: pokud je dvojice platná, odpoví $b' = 0$ (jedná se o PRF), v opačném případě odpoví $b' = 1$ (jedná se o RF).

Vidíme, že $\mathcal{D}_{\mathrm{PRF}}$ běží ve stejném čase jako $\mathcal{F}_{\mathrm{MAC}_{k_1}}$, tedy $t = t_{\mathrm{PRF}} - \mathcal{O}(1)$. Pravděpodobnost, že útočník $\mathcal{A}$ padělal MAC, můžeme zapsat jako

$$\Pr[\mathcal{A} \text{ padělal}] = \Pr[\mathcal{D}_{\mathrm{PRF}} \to 0 | b = 0]$$

Dále platí, že

$$\Pr[\mathcal{D}_{\mathrm{PRF}} \to 1 | b = 0] = 1 - \Pr[\mathcal{D}_{\mathrm{PRF}} \to 0 | b = 0] = 1 - \Pr[\mathcal{A} \text{ padělal}]$$

Protože útočník dokáže padělat pouze pro $k_1 = \mathrm{PRF}_k(1)$, kde k je náhodný, máme

$$\Pr[\mathcal{D}_{\mathrm{PRF}} \to 1 | b = 1] = 1$$

Z $(t_{\mathrm{PRF}}, q_{\mathrm{PRF}}, \epsilon_{\mathrm{PRF}})$-bezpečnosti PRF vyplývá, že

$$|\Pr[\mathcal{D}_{\mathrm{PRF}} \to 1 | b = 0] - \Pr[\mathcal{D}_{\mathrm{PRF}} \to 1 | b = 1]| < \epsilon_{\mathrm{PRF}}$$

Odsud dostáváme

$$|1 - \Pr[\mathcal{A} \text{ padělal}] - 1| = |-\Pr[\mathcal{A} \text{ padělal}]| = \Pr[\mathcal{A} \text{ padělal}] < \epsilon_{\mathrm{PRF}}$$

Pokud platí druhý případ, tedy že $\mathcal{A}$ dokáže padělat MAC pro libovolný klíč, je forger $\mathcal{F}_{\mathrm{MAC}_{k_1}}$ zároveň forger proti MAC s libovolným klíčem. Pravděpodobnost, že $\mathcal{A}$ padělal MAC je tedy v tomto případě stejná jako pravděpodobnost úspěchu $\mathcal{F}_{\mathrm{MAC}_{k_1}}$, tedy musí být menší než ϵ_{MAC}. Opět máme $t = t_{\mathrm{PRF}} - \mathcal{O}(1)$.

Shrneme-li dosažené výsledky, získáváme, že ID je rovno I_0 s pravděpodobností větší než $1 - max\{\epsilon_{\mathrm{PRF}}, \epsilon_{\mathrm{MAC}}\}$. Jak jsme již uvedli dříve, pro zjednudušení použijeme pouze výsledek v souvislosti s rodinou PRF, protože předpokládáme, že bude použita i jako schéma MAC.

■

Definice 10 (Událost MATCH) *Nechť $\hat{\mathcal{S}}$ je simulátor a nechť $\mathcal{A}$ je útočník na Σ_0. Řekneme, že v běhu $\hat{\mathcal{S}}(\mathcal{A})$ nastala událost MATCH, jestliže nastala událost GUESS a platí:*

1. *Pokud bylo sezení (I_0, s_0) zvoleno útočníkem $\mathcal{A}$ jako testovací, pak sezení (R_0, s_0) je s ním shodné.*
2. *Pokud bylo sezení (R_0, s_0) zvoleno útočníkem $\mathcal{A}$ jako testovací, pak sezení (I_0, s_0) je s ním shodné.*

Lemma 6 *Za $(t_{\mathrm{DDH}}, \epsilon_{\mathrm{DDH}})$-DDH-předpokladu a za předpokladu, že PRF je $(t_{\mathrm{PRF}}, q_{\mathrm{PRF}}, \epsilon_{\mathrm{PRF}})$-bezpečná, pro každého útočníka $\mathcal{A}$ útočícího na Σ_0 omezeného časem $t = t_{\mathrm{DDH}} - \mathcal{O}(1)$ platí, že pokud nastala událost GUESS při běhu $\hat{\mathcal{S}}-\mathrm{RAND}(\mathcal{A})$, pak nastane událost MATCH při běhu $\hat{\mathcal{S}}-\mathrm{RAND}(\mathcal{A})$ s pravděpodobností větší než $1 - 2 \cdot m \cdot n \cdot \epsilon_{\mathrm{DDH}} - \epsilon_{\mathrm{PRF}}$.*

Důkaz: Sestrojíme distinguisher ${\mathcal{D}'}_{\mathrm{DDH}}$ proti $(t_{\mathrm{DDH}}, \epsilon_{\mathrm{DDH}})$-DDH-předpokladu, který bude využívat distinguisher $\mathcal{D}_{\mathrm{DDH}}$ z důkazu lemma 2 pouze s tím rozdílem, že odpoví bit 1 pouze v případě, kdy nastane ud8lost GUESS a zároveň událost MATCH. Podobně jako v důkazu lemma 2 je $t = t_{\mathrm{DDH}} - \mathcal{O}(1)$.

Pro zkrácení zápisu budeme ve zbytku důkazu psát $\hat{\mathcal{S}}-\mathrm{H}$ místo $\hat{\mathcal{S}}-\mathrm{HYBR}$, $\hat{\mathcal{S}}-\mathrm{R}$ místo $\hat{\mathcal{S}}-\mathrm{RAND}$, G při $\hat{\mathcal{S}}-\mathrm{H}$ místo GUESS při $\hat{\mathcal{S}}-\mathrm{HYBR}$, G při $\hat{\mathcal{S}}-\mathrm{R}$ místo GUESS při $\hat{\mathcal{S}}-\mathrm{RAND}$, M při $\hat{\mathcal{S}}-\mathrm{H}$ místo MATCH při $\hat{\mathcal{S}}-\mathrm{HYBR}$ a také M při $\hat{\mathcal{S}}-\mathrm{R}$ místo MATCH při $\hat{\mathcal{S}}-\mathrm{RAND}$. Zároveň pro všechny verze simulátorů budeme psát $\hat{\mathcal{S}}$ místo $\hat{\mathcal{S}}(\mathcal{A})$.

Pro distinguisher ${\mathcal{D}'}_{\mathrm{DDH}}$ pak z lemma 2 dostáváme:

$$|\Pr[{\mathcal{D}'}^{Q_0} \to 1] - \Pr[{\mathcal{D}'}^{Q_1} \to 1]| = |\Pr[\mathcal{D}' \to 1 | z = g^r] - \Pr[\mathcal{D}' \to 1 | z = g^{xy}]| =$$
$$= |\Pr[\mathcal{D}' \to 1 \text{ při } \hat{\mathcal{S}}-\mathrm{H}] - \Pr[\mathcal{D}' \to 1 \text{ při } \hat{\mathcal{S}}-\mathrm{R}]| < \epsilon_{\mathrm{DDH}}.$$

Pro pravděpodobnost události GUESS při $\hat{\mathcal{S}}-$RAND lemma 3 říká:

$\Pr[\text{G při } \hat{\mathcal{S}}-\text{R}] \geq \frac{1}{m \cdot n}$

Z lemma 4 pak máme:

$|\Pr[\text{G při } \hat{\mathcal{S}}-\text{R}] - \Pr[\text{G při } \hat{\mathcal{S}}-\text{H}]| < \epsilon_{\text{DDH}}$

Z lemma 5 pro událost MATCH platí:

$|\Pr[\text{M při } \hat{\mathcal{S}}-\text{H}|\text{G při } \hat{\mathcal{S}}-\text{H}]| > 1 - \epsilon_{\text{PRF}}$

Tuto nerovnost si pro $0 < \delta_2 < \epsilon_{\text{PRF}}$ přepíšeme na:

$|\Pr[\text{M při } \hat{\mathcal{S}}-\text{H}|\text{G při } \hat{\mathcal{S}}-\text{H}]| = 1 - \delta_2$

Dostáváme tedy:

$|\Pr[\mathcal{D}'^{Q_0} \to 1] - \Pr[\mathcal{D}'^{Q_1} \to 1]| = |\Pr[\mathcal{D}' \to 1 \text{ při } \hat{\mathcal{S}}-\text{H}] - \Pr[\mathcal{D}' \to 1 \text{ při } \hat{\mathcal{S}}-\text{R}]|$
$= |\Pr[\text{G} \wedge \text{M při } \hat{\mathcal{S}}-\text{H}] - \Pr[\text{G} \wedge \text{M při } \hat{\mathcal{S}}-\text{R}]| = \Pr[\text{M při } \hat{\mathcal{S}}-\text{H}|\text{G při } \hat{\mathcal{S}}-\text{H}] \cdot$
$\cdot \Pr[\text{G při } \hat{\mathcal{S}}-\text{H}] - \Pr[\text{M při } \hat{\mathcal{S}}-\text{R}|\text{G při } \hat{\mathcal{S}}-\text{R}] \cdot \Pr[\text{G při } \hat{\mathcal{S}}-\text{R}] = (*)$

(1) Nejprve vyšetříme případ, kdy $\Pr[\text{G při } \hat{\mathcal{S}}-\text{H}] = \Pr[\text{G při } \hat{\mathcal{S}}-\text{R}] - \delta_1$ pro $0 < \delta_1 < \epsilon_{\text{DDH}}$. Pak dostaneme:

$(*) = |\Pr[\text{M při } \hat{\mathcal{S}}-\text{H}|\text{G při } \hat{\mathcal{S}}-\text{H}] \cdot \Pr[\text{G při } \hat{\mathcal{S}}-\text{R}] \cdot (1 - \frac{\delta_1}{\Pr[\text{G při } \hat{\mathcal{S}}-\text{R}]}) -$
$- \Pr[\text{M při } \hat{\mathcal{S}}-\text{R}|\text{G při } \hat{\mathcal{S}}-\text{R}] \cdot \Pr[\text{G při } \hat{\mathcal{S}}-\text{R}]| = |\Pr[\text{G při } \hat{\mathcal{S}}-\text{R}] \cdot$
$\cdot ((1 - \frac{\delta_1}{\Pr[\text{G při } \hat{\mathcal{S}}-\text{R}]}) \cdot (1-\delta_2)) - \Pr[\text{M při } \hat{\mathcal{S}}-\text{R}|\text{G při } \hat{\mathcal{S}}-\text{R}]| = \Pr[\text{G při } \hat{\mathcal{S}}-\text{R}] \cdot$
$\cdot |(1 - \frac{\delta_1}{\Pr[\text{G při } \hat{\mathcal{S}}-\text{R}]}) \cdot (1 - \delta_2) - 1 + 1 - \Pr[\text{M při } \hat{\mathcal{S}}-\text{R}|\text{G při } \hat{\mathcal{S}}-\text{R}]| =$
$= \Pr[\text{G při } \hat{\mathcal{S}}-\text{R}] \cdot |(1 - \frac{\delta_1}{\Pr[\text{G při } \hat{\mathcal{S}}-\text{R}]}) \cdot (1-\delta_2) - 1 + \Pr[\neg\text{M při } \hat{\mathcal{S}}-\text{R}|\text{G při } \hat{\mathcal{S}}-\text{R}]|$

(1.a) nechť $\Pr[\neg\text{M při } \hat{\mathcal{S}}-\text{R}|\text{G při } \hat{\mathcal{S}}-\text{R}] \leq |(1 - \frac{\delta_1}{\Pr[\text{G při } \hat{\mathcal{S}}-\text{R}]} \cdot (1 - \delta_2) - 1)|$,
pak $\Pr[\neg\text{M při } \hat{\mathcal{S}}-\text{R}|\text{G při } \hat{\mathcal{S}}-\text{R}] \leq |1 - \delta_2 + \frac{\delta_1(\delta_2+1)}{\Pr[\text{G při } \hat{\mathcal{S}}-\text{R}]} - 1| =$
$= \delta_2 + \frac{\delta_1(1-\delta_2)}{\Pr[\text{G při } \hat{\mathcal{S}}-\text{R}]} \leq \delta_2 + \frac{\delta_1(1-\delta_2)}{\frac{1}{m \cdot n}} = \delta_2 + m \cdot n \cdot \delta_1(1 - \delta_2) <$
$< \delta_2 + m \cdot n \cdot \delta_1 < \epsilon_{\text{PRF}} + m \cdot n \cdot \epsilon_{\text{DDH}}$

(1.b) nechť $\Pr[\neg\text{M při } \hat{\mathcal{S}}-\text{R}|\text{G při } \hat{\mathcal{S}}-\text{R}] > |(1 - \frac{\delta_1}{\Pr[\text{G při } \hat{\mathcal{S}}-\text{R}]} \cdot (1-\delta_2) - 1)|$, pak

$\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]\cdot|(1-\frac{\delta_1}{\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]})\cdot(1-\delta_2)-1+\Pr[\neg\text{M při } \hat{\mathcal{S}}\text{−R}|\text{G při } \hat{\mathcal{S}}\text{−R}]|$
$= \Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]\cdot((1-\frac{\delta_1}{\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]})\cdot(1-\delta_2)-1+\Pr[\neg\text{M při } \hat{\mathcal{S}}\text{−R}|\text{G při } \hat{\mathcal{S}}\text{−R}])$
$\geq \frac{1}{m\cdot n}\cdot((1-\frac{\delta_1}{\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]})\cdot(1-\delta_2)-1+\Pr[\neg\text{M při } \hat{\mathcal{S}}\text{−R}|\text{G při } \hat{\mathcal{S}}\text{−R}])$

Odsud dostáváme nerovnost:
$(m\cdot n)\cdot|\Pr[\mathcal{D}'^{Q_0}\to 1]-\Pr[\mathcal{D}'^{Q_1}\to 1]|\geq$
$\geq \frac{\delta_1(\delta_2-1)}{\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]}-\delta_2+\Pr[\neg\text{M při } \hat{\mathcal{S}}\text{−R}|\text{G při } \hat{\mathcal{S}}\text{−R}]$ a z té vyplývá, že:
$\Pr[\neg\text{M při } \hat{\mathcal{S}}\text{−R}|\text{G při } \hat{\mathcal{S}}\text{−R}]<m\cdot n\cdot\epsilon_{\text{DDH}}+\delta_2+\frac{\delta_1(1-\delta_2)}{\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]}<$
$<m\cdot n\cdot\epsilon_{\text{DDH}}+\epsilon_{\text{PRF}}+\frac{\delta_1(1-\delta_2)}{\frac{1}{m\cdot n}}=(m\cdot n)\cdot(\epsilon_{\text{DDH}}+\delta_1(1-\delta_2))+\epsilon_{\text{PRF}}<$
$<2\cdot m\cdot n\cdot\epsilon_{\text{DDH}}+\epsilon_{\text{PRF}}$

(2) Pokud $\Pr[\text{G při } \hat{\mathcal{S}}\text{−H}]=\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]+\delta_1$ pro $0<\delta_1<\epsilon_{\text{DDH}}$, pak

$(*)=|\Pr[\text{M při } \hat{\mathcal{S}}\text{−H}|\text{G při } \hat{\mathcal{S}}\text{−H}]\cdot\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]\cdot(1+\frac{\delta_1}{\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]})-$
$-\Pr[\text{M při } \hat{\mathcal{S}}\text{−R}|\text{G při } \hat{\mathcal{S}}\text{−R}]\cdot\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]|=|\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]\cdot$
$\cdot(\Pr[\text{M při } \hat{\mathcal{S}}\text{−H}|\text{G při } \hat{\mathcal{S}}\text{−H}]\cdot(1+\frac{\delta_1}{\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]}))-\Pr[\text{M při } \hat{\mathcal{S}}\text{−R}|\text{G při } \hat{\mathcal{S}}\text{−R}]|$
$=\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]\cdot|(1-\delta_2)\cdot(1+\frac{\delta_1}{\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]})-1+\Pr[\neg\text{M při } \hat{\mathcal{S}}\text{−R}|\text{G při } \hat{\mathcal{S}}\text{−R}]|$
$=\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]\cdot|(1-\delta_2+\frac{\delta_1(1-\delta_2)}{\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]})-1+\Pr[\neg\text{M při } \hat{\mathcal{S}}\text{−R}|\text{G při } \hat{\mathcal{S}}\text{−R}]|$

(2.a) nechť $\Pr[\neg\text{M při } \hat{\mathcal{S}}\text{−R}|\text{G při } \hat{\mathcal{S}}\text{−R}]\leq|(\frac{\delta_1(1-\delta 2)}{\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]}-\delta_2|$, pak

(2.a.a) pokud $\frac{\delta_1(1-\delta 2)}{\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]}-\delta_2\leq 0$:
$\Pr[\neg\text{M při } \hat{\mathcal{S}}\text{−R}|\text{G při } \hat{\mathcal{S}}\text{−R}]\leq\delta_2-\frac{\delta_1(1-\delta 2)}{\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]}<\delta_2<\epsilon_{\text{PRF}}$

(2.a.b) pokud $\frac{\delta_1()1-\delta 2}{\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]}-\delta_2>0$:
$\Pr[\neg\text{M při } \hat{\mathcal{S}}\text{−R}|\text{G při } \hat{\mathcal{S}}\text{−R}]\leq\frac{\delta_1()1-\delta 2}{\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]}-\delta_2\leq m\cdot n\cdot\delta_1(1-\delta_2)-\delta_2<$
$<m\cdot n\cdot\delta_1-\delta_2<m\cdot n\cdot\delta_1<m\cdot n\cdot\epsilon_{\text{DDH}}$

Odsud dostáváme:
$(m\cdot n)\cdot|\Pr[\mathcal{D}'^{Q_0}\to 1]-\Pr[\mathcal{D}'^{Q_1}\to 1]|\geq$
$\geq\frac{\delta_1(1-\delta_2)}{\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]}-\delta_2+\Pr[\neg\text{M při } \hat{\mathcal{S}}\text{−R}|\text{G při } \hat{\mathcal{S}}\text{−R}]$

Odsud pak dostáváme nerovnost:
$\Pr[\neg\text{M při } \hat{\mathcal{S}}\text{−R}|\text{G při } \hat{\mathcal{S}}\text{−R}]<max\{\epsilon_{\text{PRF}},m\cdot n\cdot\epsilon_{\text{DDH}}\}$

(2.b) nechť $\Pr[\neg\text{M při } \hat{\mathcal{S}}\text{−R}|\text{G při } \hat{\mathcal{S}}\text{−R}]>|(\frac{\delta_1(1-\delta_2)}{\Pr[\text{G při } \hat{\mathcal{S}}\text{−R}]}-\delta_2|$, pak

$\Pr[\text{G při } \hat{\mathcal{S}}{-}\text{R}] \cdot |(1 - \delta_2 + \frac{\delta_1(1-\delta_2)}{\Pr[\text{G při } \hat{\mathcal{S}}-\text{R}]} - 1 + \Pr[\neg\text{M při } \hat{\mathcal{S}}{-}\text{R}|\text{G při } \hat{\mathcal{S}}{-}\text{R}]| =$
$= \Pr[\text{G při } \hat{\mathcal{S}}{-}\text{R}] \cdot (\frac{\delta_1(1-\delta_2)}{\Pr[\text{G při } \hat{\mathcal{S}}-\text{R}]} - \delta_2 + \Pr[\neg\text{M při } \hat{\mathcal{S}}{-}\text{R}|\text{G při } \hat{\mathcal{S}}{-}\text{R}]) \geq$
$\geq \frac{1}{m \cdot n} \cdot ((\frac{\delta_1(1-\delta_2)}{\Pr[\text{G při } \hat{\mathcal{S}}-\text{R}]} - \delta_2 + \Pr[\neg\text{M při } \hat{\mathcal{S}}{-}\text{R}|\text{G při } \hat{\mathcal{S}}{-}\text{R}])$

Odsud pak dostáváme nerovnost: $\Pr[\neg\text{M při } \hat{\mathcal{S}}{-}\text{R}|\text{G při } \hat{\mathcal{S}}{-}\text{R}]$
$< m \cdot n \cdot \epsilon_{\text{DDH}} + \delta_2 - \frac{\delta_1(1-\delta_2)}{\Pr[\text{G při } \hat{\mathcal{S}}-\text{R}]} < m \cdot n \cdot \epsilon_{\text{DDH}} + \epsilon_{\text{PRF}}$

Nakonec tedy ze všech těchto vztahů dostáváme, že při běhu $\hat{\mathcal{S}}{-}\text{RAND}(\mathcal{A})$ je pravděpodobnost, že pokud nastala událost GUESS, nastala zároveň událost MATCH, větší než $1 - 2 \cdot m \cdot n \cdot \epsilon_{\text{DDH}} - \epsilon_{\text{PRF}}$.

■

Lemma 7 *Za $(t_{\text{DDH}}, \epsilon_{\text{DDH}})$-DDH-předpokladu a za předpokladu, že PRF je $(t_{\text{PRF}}, q_{\text{PRF}}, \epsilon_{\text{PRF}})$-bezpečná, pro každého útočníka $\mathcal{A}$ útočícího na Σ_0 omezeného časem $t = t_{\text{DDH}} - \mathcal{O}(1)$ platí s pravděpodobností větší než $1 - 2 \cdot m \cdot n \cdot \epsilon_{\text{DDH}} - \epsilon_{\text{PRF}}$, že $\text{Pr}_{\text{RAND}}(\mathcal{A}) = \Pr[\hat{\mathcal{S}}{-}\text{RAND}(\mathcal{A}) \to 1|\text{GUESS}]$.*

Důkaz: Při běhu $\hat{\mathcal{S}}{-}\text{RAND}$ platí, že simulátor volí hodnotu k_0 náhodně. Předpokládáme, že nastala událost GUESS, tedy nedošlo při běhu $\hat{\mathcal{S}}{-}\text{RAND}$ k přerušení a útočník $\mathcal{A}$ se v simulaci $\hat{\mathcal{S}}{-}\text{RAND}$ chová totožně, jako by se choval v běžném běhu protokolu. Jediným rozdílem mezi běžným během a simulací $\hat{\mathcal{S}}{-}\text{RAND}$ je hodnota session key k_0, která je tajným výstupem sezení (I_0, s_0) a (R_0, s_0). Ta je v případě běžného běhu protokolu rovna skutečnému session key.

BÚNO nechť $\mathcal{A}$ zvolil jako testovací sezení (I_0, s_0). Pak dle lemmatu 6 je sezení (R_0, s_0) shodné s (I_0, s_0) s pravděpodobností větší než $1-2\cdot m\cdot n\cdot \epsilon_{\text{DDH}}-\epsilon_{\text{PRF}}$. Protože $\mathcal{A}$ nesmí použít útoky session exposure na testovací sezení, ale ani na sezení s ním shodné, útočník $\mathcal{A}$ nebude mít s pravděpodobností větší než $1 - 2 \cdot m \cdot n \cdot \epsilon_{\text{DDH}} - \epsilon_{\text{PRF}}$ hodnotu k_0 k dispozici a jeho pohled nebude touto hodnotou ovlivněn.

Odsud dostáváme, že s pravděpodobností větší než $1-2\cdot m\cdot n\cdot \epsilon_{\text{DDH}}-\epsilon_{\text{PRF}}$:

$\Pr[\hat{\mathcal{S}}{-}\text{RAND}(\mathcal{A}) \to 1|\text{GUESS}] = \Pr[\mathcal{A} \to 1 \text{ při } \hat{\mathcal{S}}{-}\text{RAND}|\text{GUESS} =$
$= \Pr[\mathcal{A} \to 1 \text{ při běžném běhu, když dostane náhodnou hodnotu}] =$
$= \text{Pr}_{\text{RAND}}(\mathcal{A})$.

■

Lemma 8 *Za* $(t_{\mathrm{DDH}}, \epsilon_{\mathrm{DDH}})$*-DDH-předpokladu a za předpokladu, že PRF je* $(t_{\mathrm{PRF}}, q_{\mathrm{PRF}}, \epsilon_{\mathrm{PRF}})$*-bezpečná, pro každého útočníka* $\mathcal{A}$ *útočícího na* Σ_0 *omezeného časem* $t = t_{\mathrm{DDH}} - \mathcal{O}(1)$ *platí s pravděpodobností větší než* $1 - 2 \cdot m \cdot n \cdot \epsilon_{\mathrm{DDH}} - \epsilon_{\mathrm{PRF}}$*, že* $\mathrm{Pr}_{\mathrm{REAL}}(\mathcal{A}) = \Pr[\hat{\mathcal{S}}{-}\mathrm{REAL}(\mathcal{A}) \to 1 | \mathrm{GUESS}]$.

■

Poznámka K důkazu tohoto lemmatu je třeba použít analogie lemmat 2 až 6 pro $\hat{\mathcal{S}}{-}$REAL místo $\hat{\mathcal{S}}{-}$RAND a pro $\hat{\mathcal{S}}{-}$RPRF místo $\hat{\mathcal{S}}{-}$HYBR. Uvedeme zde pouze jejich znění bez důkazů. Ty jsou totožné s důkazy původních verzí pouze s tím rozdílem, že distinguisher $\mathcal{D}_{\mathrm{DDH}}$ z důkazu lemmat 2 a 3 použije simulátor $\hat{\mathcal{S}}{-}$REAL místo $\hat{\mathcal{S}}{-}$RAND.

Lemma 9 *Za* $(t_{\mathrm{DDH}}, \epsilon_{\mathrm{DDH}})$*-DDH-předpokladu pro použitou grupu pro každého útočníka* $\mathcal{A}$ *útočícího na* Σ_0 *a omezeného výpočetním časem* $t = t_{\mathrm{DDH}} - \mathcal{O}(1)$ *platí:*

$$\hat{\mathcal{S}}{-}\mathrm{REAL}(\mathcal{A}) \approx_{\epsilon_{\mathrm{DDH}}} \hat{\mathcal{S}}{-}\mathrm{RPRF}(\mathcal{A})$$

■

Lemma 10 *Pro libovolného útočníka* $\mathcal{A}$ *útočícího na* Σ_0 *je pravděpodobnost události GUESS při běhu simulátoru* $\hat{\mathcal{S}}{-}\mathrm{REAL}(\mathcal{A})$ *alespoň* $\frac{1}{mn}$*, kde* m *je apriorní horní mez počtu sezení iniciovaných útočníkem* $\mathcal{A}$ *a* n *je počet účastníků protokolu.*

■

Lemma 11 *Za* $(t_{\mathrm{DDH}}, \epsilon_{\mathrm{DDH}})$*-DDH-předpokladu pro použitou grupu pro každého útočníka* $\mathcal{A}$ *útočícího na* Σ_0 *a omezeného výpočetním časem* $t = t_{\mathrm{DDH}} - \mathcal{O}(1)$ *platí:*

$$\Big|\Pr[\mathrm{GUESS}\ \textit{při}\ \hat{\mathcal{S}}{-}\mathrm{REAL}(\mathcal{A})] - \Pr[\mathrm{GUESS}\ \textit{při}\ \hat{\mathcal{S}}{-}\mathrm{RPRF}(\mathcal{A})]\Big| < \epsilon_{\mathrm{DDH}}$$

■

Lemma 12 *Za předpokladu* $(t_{\mathrm{PRF}}, q_{\mathrm{PRF}}, \epsilon_{\mathrm{PRF}})$*-bezpečnosti rodiny PRF pro každého útočníka* $\mathcal{A}$ *útočícího na* Σ_0 *a omezeného časem* $t = t_{\mathrm{PRF}} - \mathcal{O}(1)$ *platí, že pokud nastala událost GUESS při běhu* $\hat{\mathcal{S}}{-}\mathrm{RPRF}(\mathcal{A})$*, pak s pravděpodobností větší než* $1 - \epsilon_{\mathrm{PRF}}$ *platí:*

1. *Pokud bylo sezení* (I_0, s_0) *zvoleno útočníkem* $\mathcal{A}$ *jako testovací, pak sezení* (R_0, s_0) *je s ním shodné.*

2. *Pokud bylo sezení* (R_0, s_0) *zvoleno útočníkem* $\mathcal{A}$ *jako testovací, pak sezení* (I_0, s_0) *je s ním shodné.*

■

Lemma 13 *Za* $(t_{\mathrm{DDH}}, \epsilon_{\mathrm{DDH}})$*-DDH-předpokladu a za předpokladu, že PRF je* $(t_{\mathrm{PRF}}, q_{\mathrm{PRF}}, \epsilon_{\mathrm{PRF}})$*-bezpečná, pro každého útočníka* $\mathcal{A}$ *útočícího na* Σ_0 *omezeného časem* $t = t_{\mathrm{DDH}} - \mathcal{O}(1)$ *platí, že pokud nastala událost GUESS při běhu* $\hat{\mathcal{S}}-\mathrm{REAL}(\mathcal{A})$*, pak nastane událost MATCH při běhu* $\hat{\mathcal{S}}-\mathrm{REAL}(\mathcal{A})$ *s pravděpodobností větší než* $1 - 2 \cdot m \cdot n \cdot \epsilon_{\mathrm{DDH}} - \epsilon_{\mathrm{PRF}}$*.*

■

Důkaz lemmatu 8 Použijeme lemmata 2 až 6 a podobně jako v důkazu lemmatu 7 dostaneme, že s pravděpodobností větší než $1-2\cdot m\cdot n\cdot\epsilon_{\mathrm{DDH}}-\epsilon_{\mathrm{PRF}}$:

$\Pr[\hat{\mathcal{S}}-\mathrm{REAL}(\mathcal{A}) \to 1|\mathrm{GUESS}] = \Pr[\mathcal{A} \to 1 \text{ při } \hat{\mathcal{S}}-\mathrm{REAL}|\mathrm{GUESS}] =$
$= \Pr[\mathcal{A} \to 1 \text{ při běžném běhu, dostane-li skutečný session key}] =$
$= \mathrm{Pr}_{\mathrm{RAND}}(\mathcal{A})$.

■

Lemma 14 *Za* $(t_{\mathrm{DDH}}, \epsilon_{\mathrm{DDH}})$*-DDH-předpokladu a za předpokladu, že PRF je* $(t_{\mathrm{PRF}}, q_{\mathrm{PRF}}, \epsilon_{\mathrm{PRF}})$*-bezpečná, pro každého útočníka* $\mathcal{A}$ *útočícího na* Σ_0 *omezeného časem* $t = max\{t_{\mathrm{DDH}}, t_{\mathrm{PRF}}\} - \mathcal{O}(1)$ *pro* $\epsilon' = 2 \cdot (\epsilon_{\mathrm{DDH}} + \epsilon_{\mathrm{PRF}})$ *platí*

$$\hat{\mathcal{S}}-\mathrm{REAL}(\mathcal{A}) \approx_{\epsilon'} \hat{\mathcal{S}}-\mathrm{RAND}(\mathcal{A})$$

Důkaz: Dle lemmatu 2 pro každého $\mathcal{A}$ omezeného časem $t_{\mathrm{DDH}} - \mathcal{O}(1)$ platí:

$$|\Pr[\hat{\mathcal{S}}-\mathrm{RAND}(\mathcal{A}) \to 1] - \Pr[\hat{\mathcal{S}}-\mathrm{HYBR}(\mathcal{A}) \to 1]| < \epsilon_{\mathrm{DDH}}$$

Podobně dle lemmatu 9 platí:

$$|\Pr[\hat{\mathcal{S}}-\mathrm{REAL}(\mathcal{A}) \to 1] - \Pr[\hat{\mathcal{S}}-\mathrm{RPRF}(\mathcal{A}) \to 1]| < \epsilon_{\mathrm{DDH}}$$

Dále ukážeme, že pro tohoto útočníka $\mathcal{A}$ platí:

$$|\Pr[\hat{\mathcal{S}}-\mathrm{RPRF}(\mathcal{A}) \to 1] - \Pr[\hat{\mathcal{S}}-\mathrm{ALLR}(\mathcal{A}) \to 1]| < \epsilon_{\mathrm{PRF}}$$

Sestrojíme distinguisher $\mathcal{D}_{\mathrm{PRF}}$ proti rodině PRF který bude mít k dispozici orákulum $\mathcal{O}^{\mathcal{F}^b}(\cdot)$ a který bude pracovat na základě simulátoru $\hat{\mathcal{S}}-\mathrm{RPRF}$ s tím rozdílem, že pro výpočet hodnot k_1 a k_0 použije orákulum $\mathcal{O}^{\mathcal{F}^b}(\cdot)$. Je zřejmé,

že pokud je orákulum funkce z PRF ($b = 0$), provádí distinguisher přesně simulaci $\hat{\mathcal{S}}-$RPRF.

Pokud je naopak orákulum skutečně náhodnou funkcí ($b = 1$), provádí simulaci $\hat{\mathcal{S}}-$ALLR. Odsud dostáváme, že:

$$|\Pr[\hat{\mathcal{S}}-\mathrm{RPRF}(\mathcal{A}) \to 1] - \Pr[\hat{\mathcal{S}}-\mathrm{ALLR}(\mathcal{A}) \to 1]| = |\Pr[\mathcal{D}^{\mathcal{O}^{\mathcal{F}^b}} \to 1|b = 0]- \\ - \Pr[\mathcal{D}^{\mathcal{O}^{\mathcal{F}^b}} \to 1|b = 1]|$$

Protože ale předpokládáme, že PRF je $(t_{\mathrm{PRF}}, q_{\mathrm{PRF}}, \epsilon_{\mathrm{PRF}})$-bezpečná, musí platit, že pro každého útočníka $\mathcal{A}$ je

$$|\Pr[\hat{\mathcal{S}}-\mathrm{RPRF}(\mathcal{A}) \to 1] - \Pr[\hat{\mathcal{S}}-\mathrm{ALLR}(\mathcal{A}) \to 1]| < \epsilon_{\mathrm{PRF}}$$

Z popisu distinguishera $\mathcal{D}_{\mathrm{PRF}}$ ihned dostáváme, že $t = \mathrm{PRF} - \mathcal{O}(1)$. Nyní už zbývá jen ukázat:

$$|\Pr[\hat{\mathcal{S}}-\mathrm{ALLR}(\mathcal{A}) \to 1] - \Pr[\hat{\mathcal{S}}-\mathrm{HYBR}(\mathcal{A}) \to 1]| < \epsilon_{\mathrm{PRF}}$$

Použijeme opět distinguisher $\mathcal{D}_{\mathrm{PRF}}$ s tím rozdílem, že tentokrát použije orákulum pouze k výpočtu hodnoty k_1. Hodnotu k_0 volí náhodně, stejně jako simulátor $\hat{\mathcal{S}}-$ALLR. Vidíme, že pokud $\mathcal{D}_{\mathrm{PRF}}$ komunikuje s pseudonáhodnou funkcí, provádí simulaci $\hat{\mathcal{S}}-$HYBR. Pokud komunikuje s náhodnou funkcí, provádí simulaci $\hat{\mathcal{S}}-$ALLR. Odsud a z $(t_{\mathrm{PRF}}, q_{\mathrm{PRF}}, \epsilon_{\mathrm{PRF}})$-bezpečnosti PRF plyne, že:

$$|\Pr[\hat{\mathcal{S}}-\mathrm{RPRF}(\mathcal{A}) \to 1] - \Pr[\hat{\mathcal{S}}-\mathrm{ALLR}(\mathcal{A}) \to 1]| = |\Pr[\mathcal{D}^{\mathcal{O}^{\mathcal{F}^b}} \to 1|b = 0]- \\ - \Pr[\mathcal{D}^{\mathcal{O}^{\mathcal{F}^b}} \to 1|b = 1]| < \epsilon_{\mathrm{PRF}}$$

Opět dostáváme $t = \mathrm{PRF} - \mathcal{O}(1)$. Použijeme trojúhelníkovou nerovnost a získáme:

$$|\Pr[\hat{\mathcal{S}}-\mathrm{RPRF}(\mathcal{A}) \to 1] - \Pr[\hat{\mathcal{S}}-\mathrm{ALLR}(\mathcal{A}) \to 1]| + |\Pr[\hat{\mathcal{S}}-\mathrm{ALLR}(\mathcal{A}) \to 1]- \\ - \Pr[\hat{\mathcal{S}}-\mathrm{HYBR}(\mathcal{A}) \to 1]| \geq |\Pr[\hat{\mathcal{S}}-\mathrm{RPRF}(\mathcal{A}) \to 1]-\Pr[\hat{\mathcal{S}}-\mathrm{ALLR}(\mathcal{A}) \to 1]+ \\ + \Pr[\hat{\mathcal{S}}-\mathrm{ALLR}(\mathcal{A}) \to 1] - \Pr[\hat{\mathcal{S}}-\mathrm{HYBR}(\mathcal{A}) \to 1]| = \\ = |\Pr[\hat{\mathcal{S}}-\mathrm{RPRF}(\mathcal{A}) \to 1] - \Pr[\hat{\mathcal{S}}-\mathrm{HYBR}(\mathcal{A}) \to 1]|$$

Odsud plyne, že:

$$|\Pr[\hat{\mathcal{S}}-\mathrm{RPRF}(\mathcal{A}) \to 1] - \Pr[\hat{\mathcal{S}}-\mathrm{HYBR}(\mathcal{A}) \to 1]| < 2 \cdot \epsilon_{\mathrm{PRF}}$$

Dále získáme:

$|\Pr[\hat{\mathcal{S}}{-}\mathrm{REAL}(\mathcal{A}) \to 1] - \Pr[\hat{\mathcal{S}}{-}\mathrm{RPRF}(\mathcal{A}) \to 1]| + |\Pr[\hat{\mathcal{S}}{-}\mathrm{RPRF}(\mathcal{A}) \to 1]-$
$- \Pr[\hat{\mathcal{S}}{-}\mathrm{HYBR}(\mathcal{A}) \to 1]| \geq |\Pr[\hat{\mathcal{S}}{-}\mathrm{REAL}(\mathcal{A}) \to 1]-\Pr[\hat{\mathcal{S}}{-}\mathrm{RPRF}(\mathcal{A}) \to 1]+$
$+ \Pr[\hat{\mathcal{S}}{-}\mathrm{RPRF}(\mathcal{A}) \to 1] - \Pr[\hat{\mathcal{S}}{-}\mathrm{HYBR}(\mathcal{A}) \to 1]| =$
$= |\Pr[\hat{\mathcal{S}}{-}\mathrm{REAL}(\mathcal{A}) \to 1] - \Pr[\hat{\mathcal{S}}{-}\mathrm{HYBR}(\mathcal{A}) \to 1]|$

Odsud plyne, že:

$$|\Pr[\hat{\mathcal{S}}{-}\mathrm{REAL}(\mathcal{A}) \to 1] - \Pr[\hat{\mathcal{S}}{-}\mathrm{HYBR}(\mathcal{A}) \to 1]| < 2 \cdot \epsilon_{\mathrm{PRF}} + \epsilon_{\mathrm{DDH}}$$

Nakonec získáme:

$|\Pr[\hat{\mathcal{S}}{-}\mathrm{REAL}(\mathcal{A}) \to 1] - \Pr[\hat{\mathcal{S}}{-}\mathrm{HYBR}(\mathcal{A}) \to 1]| + |\Pr[\hat{\mathcal{S}}{-}\mathrm{HYBR}(\mathcal{A}) \to 1]-$
$- \Pr[\hat{\mathcal{S}}{-}\mathrm{RAND}(\mathcal{A}) \to 1]| \geq$
$\geq |\Pr[\hat{\mathcal{S}}{-}\mathrm{REAL}(\mathcal{A}) \to 1] - \Pr[\hat{\mathcal{S}}{-}\mathrm{HYBR}(\mathcal{A}) \to 1] + \Pr[\hat{\mathcal{S}}{-}\mathrm{HYBR}(\mathcal{A}) \to 1]-$
$- \Pr[\hat{\mathcal{S}}{-}\mathrm{RAND}(\mathcal{A}) \to 1]| =$
$= |\Pr[\hat{\mathcal{S}}{-}\mathrm{REAL}(\mathcal{A}) \to 1] - \Pr[\hat{\mathcal{S}}{-}\mathrm{RAND}(\mathcal{A}) \to 1]|$

Odsud plyne, že:

$$|\Pr[\hat{\mathcal{S}}{-}\mathrm{REAL}(\mathcal{A}) \to 1] - \Pr[\hat{\mathcal{S}}{-}\mathrm{RAND}(\mathcal{A}) \to 1]| < 2 \cdot (\epsilon_{\mathrm{PRF}} + \epsilon_{\mathrm{DDH}})$$

■

Poznámka Nyní už máme vše potřebné a můžeme přistoupit k samotnému důkazu vlastnosti 2 věty 1, tedy tak dokončit celý důkaz (t, q, δ, ϵ)-SK-bezpečnosti protokolu Σ_0.

Důkaz vlastnosti 2: Nejdříve ze všeho si vyjádříme pravděpodobnosti $\Pr[\hat{\mathcal{S}}{-}\mathrm{RAND}(\mathcal{A}) \to 1]$ a $\Pr[\hat{\mathcal{S}}{-}\mathrm{REAL}(\mathcal{A}) \to 1]$:

První pravděpodobnost můžeme rozepsat jako:

$\Pr[\hat{\mathcal{S}}{-}\mathrm{RAND}(\mathcal{A}) \to 1] =$
$= \Pr[\hat{\mathcal{S}}{-}\mathrm{RAND}(\mathcal{A}) \to 1|\mathrm{GUESS}] \cdot \Pr[\mathrm{GUESS\ při\ } \hat{\mathcal{S}}{-}\mathrm{RAND}]+$
$+ \Pr[\hat{\mathcal{S}}{-}\mathrm{RAND}(\mathcal{A}) \to 1|\neg\mathrm{GUESS}] \cdot \Pr[\neg\mathrm{GUESS\ při\ } \hat{\mathcal{S}}{-}\mathrm{RAND}] =$
$= \Pr[\hat{\mathcal{S}}{-}\mathrm{RAND}(\mathcal{A}) \to 1|\mathrm{GUESS}] \cdot \Pr[\mathrm{GUESS\ při\ } \hat{\mathcal{S}}{-}\mathrm{RAND}] \geq$
$\geq \Pr[\hat{\mathcal{S}}{-}\mathrm{RAND}(\mathcal{A}) \to 1|\mathrm{GUESS}] \cdot \frac{1}{m \cdot n}$

Druhou pravděpodobnost můžeme rozepsat jako:

$\Pr[\hat{\mathcal{S}}{-}\mathrm{REAL}(\mathcal{A}) \to 1] =$

$= \Pr[\hat{\mathcal{S}}{-}\mathrm{REAL}(\mathcal{A}) \to 1|\mathrm{GUESS}] \cdot \Pr[\mathrm{GUESS\ při}\ \hat{\mathcal{S}}{-}\mathrm{REAL}] +$
$+ \Pr[\hat{\mathcal{S}}{-}\mathrm{REAL}(\mathcal{A}) \to 1|\neg\mathrm{GUESS}] \cdot \Pr[\neg\mathrm{GUESS\ při}\ \hat{\mathcal{S}}{-}\mathrm{REAL}] =$
$= \Pr[\hat{\mathcal{S}}{-}\mathrm{REAL}(\mathcal{A}) \to 1|\mathrm{GUESS}] \cdot \Pr[\mathrm{GUESS\ při}\ \hat{\mathcal{S}}{-}\mathrm{REAL}] \geq$
$\geq \Pr[\hat{\mathcal{S}}{-}\mathrm{REAL}(\mathcal{A}) \to 1|\mathrm{GUESS}] \cdot \frac{1}{m \cdot n}$

Vycházeli jsme při tom z lemmat 3 a 10 a použili jsme fakt, že pokud nenastane událost GUESS, simulace je přerušena a simulátor vždy odpoví bit 0. Nyní si popíšeme dvě situace, které mohou při útoku $\mathcal{A}$ na testovací sezení nastat.

1. $\mathcal{A}$ při útoku zvolil jako testovací sezení (I_0, s_0), které se dokončilo s peerem R_0, a sezení (R_0, s_0) je s ním shodné. Pak dle lemmat 7 a 8 platí, že:

$$\Pr_{\mathrm{RAND}}(\mathcal{A}) = \Pr[\hat{\mathcal{S}}{-}\mathrm{RAND}(\mathcal{A}) \to 1|\mathrm{GUESS}]$$

$$\Pr_{\mathrm{REAL}}(\mathcal{A}) = \Pr[\hat{\mathcal{S}}{-}\mathrm{REAL}(\mathcal{A}) \to 1|\mathrm{GUESS}]$$

   V tomto případě můžeme psát:

$\mathbf{Adv}^{\mathrm{test}}_{\Sigma_0}(\mathcal{A}) = |\Pr_{\mathrm{REAL}(\mathcal{A})} - \Pr_{\mathrm{RAND}(\mathcal{A})}| =$
$= |\Pr[\hat{\mathcal{S}}{-}\mathrm{REAL}(\mathcal{A}) \to 1|\mathrm{GUESS}] - \Pr[\hat{\mathcal{S}}{-}\mathrm{RAND}(\mathcal{A}) \to 1|\mathrm{GUESS}]| \leq$
$\leq m \cdot n \cdot |\Pr[\hat{\mathcal{S}}{-}\mathrm{REAL}(\mathcal{A}) \to 1] - \Pr[\hat{\mathcal{S}}{-}\mathrm{RAND}(\mathcal{A}) \to 1]| <$
$< 2 \cdot m \cdot n \cdot (\epsilon_{\mathrm{DDH}} + \epsilon_{\mathrm{PRF}})$

   Tedy pravděpodobnost úspěchu útočníka $\mathcal{A}$ je v tomto případě rovna

$$\frac{1}{2} + \alpha \text{ pro } \alpha < 2 \cdot m \cdot n \cdot (\epsilon_{\mathrm{DDH}} + \epsilon_{\mathrm{PRF}})$$

2. $\mathcal{A}$ při útoku zvolil jako testovací sezení (I_0, s_0), které se dokončilo s peerem R_0, ale sezení (R_0, s_0) s ním není shodné. Jinými slovy útočníkovi se podařilo zaútočit na (R_0, s_0) takovým způsobem, aby jeho veřejný výstup byl (R_0, s_0, ID), kde $\mathrm{ID} \neq I_0$. V tomto případě může útočník použít na (R_0, s_0) session exposure, získat hodnotu session key a uspět v útoku na testovací sezení s pravděpodobností 1.

   Použitím stejného argumentu jako v důkazu lemma 5 (pokud by útočník uměl padělat MAC_{k_1}, můžeme sestrojit forger proti MAC_{k_1}, na jehož základě můžeme sestrojit distinguisher proti PRF) získáváme, že toto se útočníkovi podaří s pravděpodobností menší než ϵ_{PRF}. Z výše uvedeného tedy vyplývá, že pravděpodobnost situace 2 je rovna $\beta < \epsilon_{\mathrm{PRF}}$ a pravděpodobnost situace 1 je rovna $1 - \beta$.

Celkově tedy získáváme, že:

$\Pr[\mathcal{A} \text{ uspěl}] =$
$= \Pr[\mathcal{A} \text{ uspěl} | (R_0, s_0) \text{ shodné s } (I_0, s_0)] \cdot \Pr[(R_0, s_0) \text{ shodné s } (I_0, s_0)] +$
$+ \Pr[\mathcal{A} \text{ uspěl} | (R_0, s_0) \text{ není shodné s } (I_0, s_0)] \cdot$
$\cdot \Pr[(R_0, s_0) \text{ není shodné s } (I_0, s_0)] = (\frac{1}{2} + \alpha) \cdot (1 - \beta) + \beta =$
$= \frac{1}{2} + \alpha - \frac{\beta}{2} - \beta \cdot \alpha + \beta < \frac{1}{2} + \frac{\epsilon_{\text{PRF}}}{2} + 2 \cdot m \cdot n \cdot (\epsilon_{\text{DDH}} + \epsilon_{\text{PRF}})$.

■

Kapitola 7

Závěr

V závěru shrneme konkrétní výsledky, které jsme o protokolu dokázali. Také popíšeme, jakým způsobem je zajištěn přechod k protokolu IKE, a jak je tím bezpečnost ovlivněna. Již nebudeme uvádět konkrétní hodnoty, ale obecné argumenty zachování SK bezpečnosti při mírné úpravě kroků protokolu. Podrobně jsou tyto argumenty popsány v [2] a konkrétní důkaz by probíhal analogicky, jako to bylo v důkazu bezpečnosti Σ_0.

7.1 Shrnutí výsledků

Ačkoliv jsme dosažené výsledky shrnuli již formulací věty 1, neuškodí si je zde přehledně uvést znovu. Pro útočníka na protokol Σ_0 jsme dokázali, že omezení jeho výpočetního času se v podstatě neliší od omezení, která vyplývají z použitých primitivů. Získali jsme hodnotu $t = max\{t_{\mathrm{DDH}}, t_{\mathrm{PRF}}, t_{\mathrm{SIG}}\} - \mathcal{O}(1)$.

Jako hodnotu advantage tohoto účastníka jsme odvodili $\epsilon = \epsilon_{\mathrm{PRF}} \cdot (\frac{1}{2} + 2mn) + \epsilon_{\mathrm{DDH}} \cdot 2mn$. Pomocí této hodnoty můžeme vyjádřit, s jakou úspěšností dokáže útočník odvozovat nějaké dodatečné informace o tajném klíči. Zde je třeba zmínit, jak velkými faktory jsou ve skutečnosti hodnoty m a n. Jedná se o apriorní horní mez počtu sezení iniciovaných útočníkem v běhu protokolu, tedy toto číslo není řádově velké, a o počet účastníků protokolu. Protože předpokládáme, že útočník musí každého účastníka iniciovat, tedy to udělá pouze v případě, kdy je to pro něj výhodné, můžeme říci, že toto číslo také není řádově velké. Nejedná se o počet všech možných účastníků, ale o počet, který útočník použije při svém útoku.

Třetí dosaženou hodnotou je $\delta = 1 - \epsilon_{\mathrm{SIG}}$. Ta vyjadřuje, jak (ne)úspěšný je útočník při narušení konzistence protokolu, tedy při sabotování probíhající výměny klíče, aniž by si toho byl účastník vědom. Vidíme, že tato hodnota je přímo závislá na použité metodě ochrany autentičnosti a integrity.

7.2 Přechod k IKE

Pro získání hlavního módu protokolu IKE je třeba třeba v protokolu Σ_0 odstranit příznaky iniciátora a respondéra, které jsou v Σ_0 zahrnuty do výpočtu podpisu a MACu. Jejich přítomnosti jsme v důkazu využili pouze v lemmatu 1 a 5. Z něj snadno vidíme, že odebrání těchto příznaků bezpečnost protokolu neovlivní. Další změnou je to, že hodnota MACu se neposílá zvlášť, ale je zahrnuta do výpočtu podpisu. To situaci také nijak drasticky nemění. Stále platí, že musí být ověřen podpis i MAC. Třetí změnou je použití šifrování na poslední dvě zprávy a to jistě nevede ke snížení bezpečnosti. Připomeňme, že rychlá fáze IKE je již chráněna sdílenými klíči, tedy bezpečnost celého protokolu je závislá na bezpečnosti jeho první fáze. Tedy je na ní závislá i bezpečnost protokolu IPSec. Varianta IKEv2 se od IKEv1 liší pouze tím, že zkomprimovala dvě fáze protokolu IKEv1 do jedné (druhá fáze nastupuje až podle potřeby). Mechanismus ochrany bezpečnosti je v podstatě stále stejný. Další změnou je také implementování MACu pomocí PRF, ale toto jsme v důkazu předpokládali, tedy námi odvozené hodnoty odpovídají protokolu IKE.

Literatura

[1] Canetti, R. a Krawczyk, H. (2001). Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. Lecture Notes in Computer Science, pp.453-474. [cit. 2015-07-29]. Dostupné z: http://iacr.org/archive/eurocrypt2001/20450451.pdf.

[2] Canetti, R., Krawczyk, H. (2002). Security Analysis of IKE's Signature-based Key-Exchange Protocol [cit. 2015-07-29]. Dostupné z: http://www.iacr.org/archive/crypto2002/24420143/24420143.pdf

[3] Frankel, S. a Krishnan, S. (2011). IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap. [cit. 2015-07-29] Dostupné z: https://tools.ietf.org/html/rfc6071.

[4] Harkins, D. a Carrel, D. (1998). The Internet Key Exchange (IKE). [cit. 2015-07-29] Dostupné z: https://tools.ietf.org/html/rfc2409.

[5] Katz, J. and Lindell, Y. (2014) Introduction to Modern Cryptography, Second Edition. CRC Press, p.603 ISBN 01-360-9704-9.

[6] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P. a Kivinen, T. (2014). Internet Key Exchange Protocol Version 2 (IKEv2). [cit. 2015-07-29] Dostupné z: https://tools.ietf.org/html/rfc7296.

[7] Kozierok, CH. M. (2005) The TCP/IP Guide [online]. Verze 3.0., 27. září 2005 [cit. 2015-07-29]. Dostupné z: http://www.tcpipguide.com

[8] Krawczyk, H. (2003). SIGMA: The 'SIGn-and-MAc' Approach to Authenticated Diffie-Hellman and Its Use in the IKE Protocols. Advances in Cryptology - CRYPTO 2003, pp.400-425 [cit. 2015-07-29]. Dostupné z: http://www.iacr.org/cryptodb/archive/2003/CRYPTO/1495/1495.pdf.

[9] Maughan, D., Schertler, M., Schneider, M. a Turner, J. (1998). Internet Security Association and Key Management Protocol (ISAKMP). [cit. 2015-07-29] Dostupné z: https://tools.ietf.org/html/rfc2408.

[10] Piper, D. (1998). The Internet IP Security Domain of Interpretation for ISAKMP. [cit. 2015-07-29] Dostupné z: https://tools.ietf.org/html/rfc2407.

[11] Stallings, W. (2011). Cryptography and network security: principles and practice. 5th ed. Boston: Prentice Hall, c2011, xxiii, 719 p. ISBN 01-360-9704-9.