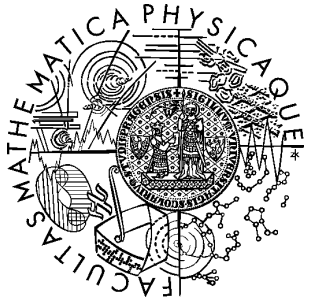
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Rudolf Barczi

Důvěryhodnost prokazatelně bezpečné kryptografie

Katedra algebry

Vedoucí diplomové práce: RNDr. Bohuslav Rudolf

Studijní program: Matematika
Studijní obor: Matematické metody informační bezpečnosti

2009

Ďakujem svojmu vedúcemu diplomovej práce za odborné vedenie, vďaka ktorému bolo možné túto prácu flexibilne smerovať až k najnovším poznatkom v oblasti preukázateľnej bezpečnosti. Základom boli mnohopočetné konzultácie, kde mi RNDr. Bohuslav Rudolf vždy ochotne, trpezlivo a so záujmom napomáhal pri štúdiu nevyhnutnému k pochopeniu danej problematiky. Taktiež som vďačný za jeho cenné rady, trefné pripomienky a inšpiratívne podnety pri samotnom písaní. V neposlednej rade ďakujem za všetok čas, ktorý venoval detailnej revízii tejto práce.

Prehlasujem, že som svoju diplomovú prácu napísal samostatne a výhradne s použitím citovaných prameňov. Súhlasím so zapožičiavaním práce a jej zverejňovaním.

V Prahe dňa 30. júla 2009 Rudolf Barczi

Obsah

Názov práce: Důvěryhodnost prokazatelně bezpečné kryptografie
Autor: Rudolf Barczi
e-mailová adresa: rudolf@barczi.net
Katedra: Katedra algebry
Vedúci práce: RNDr. Bohuslav Rudolf
e-mail vedúceho práce: b.rudolf@nbu.cz

Abstrakt: V predloženej práci študujeme problematiku preukázateľnej bezpečnosti v modele náhodného orákula a štandardnom modele na príklade kryptosystému OAEP. Na začiatku predstavíme všeobecné pojmy z oblasti asymetrickej kryptografie. V ďalšej kapitole sledujeme genézu dokazovania bezpečnosti kryptosystému RSA-OAEP v modele náhodného orákula od pôvodného kontroverzného dôkazu z roku 1994 až po korektný a technicky náročný dôkaz z roku 2004. Nasledujúca kapitola je venovaná vybraným problematickým aspektom praktickej bezpečnosti RSA-OAEP. Cieľom rozsiahlej štvrtej kapitoly je prezentácia niektorých najnovších výsledkov týkajucich sa bezpečnosti RSA-OAEP v štandardnom modele. Prvý z nich pochádza z roku 2009 a ukazuje principiálnu nemožnosť konštrukcie dôkazu bezpečnosti RSA-OAEP v zmysle CCA2. Výsledok z roku 2006 je naopak síce pozitívny (slabá neohybnosť plne inštancializovaného OAEP), ale jeho váha je diskutabilná. Záver práce obsahuje niekoľko komentárov k súčasnému stavu preukázateľnej bezpečnosti RSA-OAEP.

Kľúčové slová: preukázateľná bezpečnosť, OAEP, model náhodného orákula, štandardný model

Title: Credibility of Provable Secure Cryptography
Author: Rudolf Barczi
e-mail address: rudolf@barczi.net
Department: Department of Algebra
Supervisor: RNDr. Bohuslav Rudolf
Supervisor's e-mail address: b.rudolf@nbu.cz

Abstract: In the present work we study provable security in the random oracle model and the standard model using the OAEP cryptosystem as an example. We begin with general introduction to public-key cryptography. In the next chapter we trace the evolution of RSA-OAEP cryptosystem security proofs in the random oracle model from the original controversial proof of security from 1994 to the correct and technically challenging one from 2004. The third chapter is dedicated to the selected problematic aspects of RSA-OAEP practical security. The goal of the extensive fourth chapter is to present some of the most recent results regarding the security of RSA-OAEP in the standard model. The first result from 2009 shows the fundamental impossibility of security proof construction in the sense of CCA2. The result from 2006, despite being positive (weak non-malleability of fully-instantiated OAEP), is of an arguable significance. In the end we mention some comments on the state-of-the-art provable security of RSA-OAEP.

Keywords: provable security, OAEP, random oracle model, standard model

1 Úvod

1.1. Kryptografia s verejným kľúčom

Koncept kryptografie s verejným kľúčom je pozoruhodný vďaka svojej jednoduchosti, elegancii a najmä širokému spektru praktickej aplikovateľnosti. V tomto oddiele si ho predstavíme.

1.1.1. Základné pojmy

Pre úplnosť začneme definíciou šifrového systému.

Definícia (šifrový systém). *Šifrovým systémom* (tiež kryptosystémom) nazývame usporiadanú sedmicu (A, M, C, K, E, D, G), kde:

- A je abeceda prípustných symbolov (typicky $A = \{0, 1\}$),
- M je priestor prípustných správ, pričom správou rozumieme reťazec znakov z abecedy A,
- C je priestor prípustných šifrových textov, pričom šifrovým textom rozumieme reťazec znakov z abecedy A,
- K je priestor kľúčov, pričom platí:
  - každý prvok $e \in K$ určuje zobrazenie $E_e : M \to C$, ktoré sa nazýva šifrovacia transformácia. Zároveň je $E = \{E_e | e \in K\}$. Procesu aplikácie šifrovacej transformácie, teda výpočtu funkčnej hodnoty $E_e(m)$ pre nejaké $m \in M$, hovoríme šifrovanie.
  - každý prvok $d \in K$ určuje zobrazenie $D_d : M \to C$, ktoré sa nazýva dešifrovacia transformácia. Zároveň je $D = \{D_d | d \in K\}$. Procesu aplikácie dešifrovacej transformácie, teda výpočtu funkčnej hodnoty $D_d(c)$ pre nejaké $c \in C$, hovoríme dešifrovanie.
  - množiny E a D si vzájomne odpovedajú podľa vzťahu:

$$(\forall e \in E)(\exists d \in D)(\forall m \in M) : D_d(E_e(m)) = m.$$

- G je generátor prípustných kľúčov.

Uvedená definícia je prispôsobená kryptosystémom s verejným kľúčom (tzv. asymetrickým kryptosystémom), ktorých preukázateľnou bezpečnosťou sa v tejto práci budeme zaoberať. Problematiku bezpečnosti však táto definícia nerieši, lebo i keď je bezpečnosť neodmysliteľným aspektom každého praktického kryptosystému, jedná sa o formálne ťažšie uchopiteľnú problematiku. Preto sa ňou explicitne zaoberáme až v samostatnom oddiele 1.2.

Poznámka. Značenie $\forall^* x \in M$ bude v ďalšom texte kvantifikovať skoro všetky prvky x množiny M (tzn. existuje najviac konečne mnoho prvkov $y \in M$, ktoré uvažovaná kvantifikácia nezahrňuje).

DEFINÍCIA (zanedbateľná funkcia). Funkciu $\nu : \mathbb{N} \to [0,1]$ nazývame *zanedbateľnou*, ak platí:

$$(\forall c > 0)\ (\forall^* k \in \mathbb{N}) : \nu(k) < \frac{1}{k^c}.$$

POZNÁMKA. Značením $\nu \approx \rho$ budeme pre dve funkcie $\nu, \rho : \mathbb{N} \to [0,1]$ rozumieť, že $|\nu(k) - \rho(k)|$ je zanedbateľná funkcia.

POZNÁMKA. Veľkosťou prvku $x \in M$ budeme rozumieť počet bitov potrebných na jeho reprezentáciu v binárnej sústave, pričom i-ty najnižší bit prvku x budeme označovať ako $x[i-1]$.

DEFINÍCIA (jednosmerná funkcia). Funkciu $f : X \to Y$ nazývame *jednosmernou*, ak sú splnené nasledujúce podmienky:

- existuje pravdepodobnostný polynomiálny (vo veľkosti $x \in X$) algoritmus $\mathcal{A}$ taký, že
$$(\forall x \in X) : \mathcal{A}(x) = f(x),$$
- pre každý pravdepodobnostný polynomiálny (vo veľkosti $y \in Y$) algoritmus $\mathcal{B}$ a $\forall^* y \in Y$ je pravdepodobnosť
$$\Pr[\mathcal{B}(y) = f^{-1}(y)]$$
zanedbateľnou funkciou vo veľkosti $y \in Y$.

POZNÁMKA. Jednosmernú funkciu, ktorá je zároveň bijekciou, nazývame *jednosmerná permutácia*.

POZNÁMKA. Ak nebude uvedené inak, budeme každý algoritmus, ako aj každú entitu vykonávajúcu algoritmickú činnosť (napríklad útočník), v ďalšom texte modelovať pravdepodobnostným polynomiálnym (v dĺžke vstupu) algoritmom (Turingovým strojom). Výpočtovou nedosažiteľnosťou nejakej úlohy budeme rozumieť neexistenciu pravdepodobnostného polynomiálneho (v dĺžke vstupu) algoritmu, ktorého pravdepodobnosť úspechu (vyriešenia danej úlohy) nie je možné zhora ohraničiť nejakou zanedbateľnou funkciou veľkosti vstupu do tejto úlohy.

Predošlé definície formálne zachycujú intuitívne pojmy, ktoré sú základnými stavebnými kameňmi modernej kryptografie. Zmysel jednosmernej permutácie spočíva v jej asymetrii – na jednej strane sa funkčná hodnota na danom vstupe vždy ľahko vypočíta, no zároveň je na strane druhej skoro vždy tento proces výpočtovo nedosažiteľné invertovať, tj. nájsť k nejakej funkčnej hodnote odpovedajúci vstup.

Podobné požiadavky sú kladené aj na fungovanie šifrových systémov, kde je nutné zaistiť, aby sa každá správa dala ľahko zašifrovať a aby bolo súčasne ťažké z príslušného šifrového textu túto správu rekonštruovať. Takže nutným predpokladom bezpečnosti šifrového systému je výpočtová nedosažiteľnosť úlohy nájsť na základe znalosti e nejaké odpovedajúce d tak, aby platilo

$$(\forall m \in M) : D_d(E_e(m)) = m.$$

Naplnenie výlučne uvedeného predpokladu však ešte nezaručuje praktickú použiteľnosť šifrového systému, pretože sa môže stať, že zašifrovanú správu už nebude možné odšifrovať. To vedie k pojmu jednosmernej permutácie s padacími vrátkami.

Jednosmerná permutácia s padacími vrátkami je jednosmerná permutácia, ku ktorej navyše existuje nejaká dodatočná informácia umožňujúca túto jednosmernú permutáciu ľahko (tj. v polynomiálnom čase) invertovať. Práve dodatočná informácia býva v praxi tajným dešifrovacím kľúčom, ktorého odvodenie z verejne získateľných informácií (popis permutácie, verejný kľúč, šifrové texty, otvorené texty, atd.) je výpočtovo nedosažiteľnou úlohou. Dodatočná informácia teda svojmu vlastníkovi slúži k otvoreniu padacích vrátok, ktoré sa zašifrovaním správy jednosmernou permutáciou zatvoria. Problematikou jednosmerných permutácií s padacími vrátkami sa podrobne zaoberáme v oddiele 4.1.1.

1.1.2. Princíp fungovania

Kryptografia s verejným kľúčom je nazývaná tiež asymetrická kryptografia. Pôvod tohoto alternatívneho pomenovania spočíva vo fakte, že na šifrovanie aj dešifrovanie sa používajú odlišné kľúče. Protikladom sú symetrické šifry (napríklad AES), kde tieto kľúče splývajú. Prejdime ku konkrétnemu popisu asymetrického šifrového systému.

Nech K označuje priestor kľúčov. Označme $E = \{E_e | e \in K\}$ množinu šifrovacích transformácií a $D = \{D_d | d \in K\}$ množinu odpovedajúcich dešifrovacích transformácií. Uvažujme nejakú dvojicu asociovaných šifrovacích a dešifrovacích transformácií (E_e, D_d), teda dvojicu splňujúcu vzťah:

$$(\forall m \in M) : D_d(E_e(m)) = m.$$

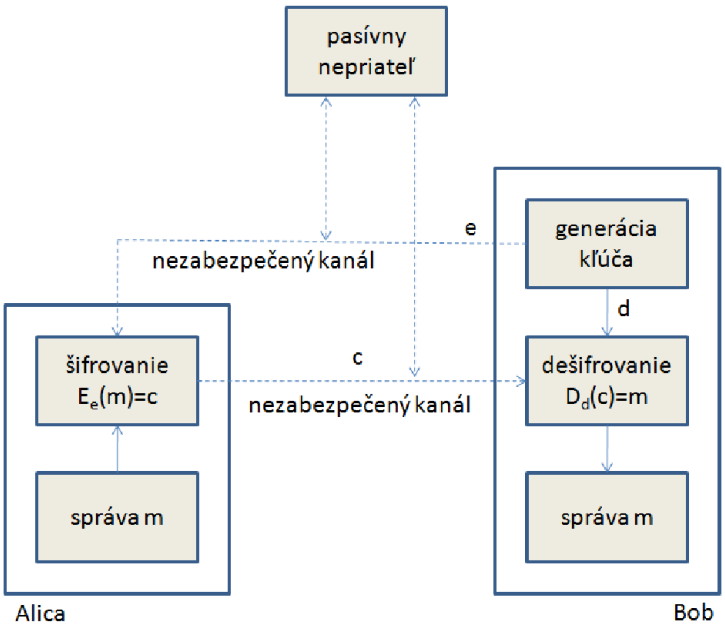
Ďalej predpokladajme, že v prípade neznalosti transformácie D_d je pre ľubovoľný náhodne zvolený šifrový text $c \in C$ výpočtovo nedosažiteľné nájsť správu $m \in M$ tak, aby platilo:

$$E_e(m) = c.$$

Uvedené predpoklady znamenajú, že zo samotnej znalosti šifrovacieho kľúča e je výpočtovo nedosažiteľné odvodiť odpovedajúci dešifrovací kľúč d. Transformácia E_e teda v popísanej situácii predstavuje jednosmernú funkciu s padacími vrátkami, pričom onými padacími vrátkami potrebnými k jej invertovaniu je práve dešifrovací kľúč d. Problémom však ostáva fakt, že o žiadnom reálnom asymetrickom šifrovom systéme nebolo ani napriek množstvu indícií dosiaľ exaktne matematicky dokázané, že jeho množina šifrovacích transformácií E je množinou jednosmerných funkcií s padacími vrátkami. Bezpečnosť šifrových systémov s verejným kľúčom je preto založená na predpoklade existencie jednosmerných funkcií s padacími vrátkami. Konštrukcia takých funkcií v praxi vychádza vždy z konkrétneho matematického problému, ktorého riešenie je všeobecne považované za výpočtovo nedosažiteľnú úlohu (napr. problém faktorizácie celých čísel, problém diskrétneho logaritmu).

Poznámka. V ďalšom texte sa budeme držať štandardnej konvencie a označovať šifrovací kľúč e ako kľúč verejný a dešifrovací kľúč d ako kľúč privátny. Zmysel tohoto pomenovania je vysvetlený nižšie.

Definícia (asymetrický šifrový systém). Šifrový systém (A, M, C, K, E, D, G) sa nazýva *asymetrický*, ak z každého asociovaného páru kľúčov (e, d), kde $e, d \in K$, je verejný kľúč e dostupný komukoľvek a privátny kľúč d ostáva v utajení u svojho vlastníka.



Obr. 1.1. Schéma šifrového systému s verejným kľúčom

Poznámka. Nezabezpečeným kanálom rozumieme taký komunikačný kanál spojujúci dve strany, v ktorom nie je tretím stranám zamedzená možnosť prebiehajúcu komunikáciu sledovať (pasívni útočníci) a ovplyvňovať (aktívni útočníci).

Za daných okolností uvažujme dve entity, Alicu a Boba, komunikáciu ktorých zachycuje Obrázok 1.1. Bob zvolí dvojicu kľúčov (e, d) a verejný kľúč pošle Alici (zverejní ho). Privátny kľúč d však uchová v tajnosti. Vďaka znalosti verejného kľúča e môže Alica následne odoslať Bobovi správu m v zašifrovanom tvare tak, že na túto správu m použije šifrovaciu transformáciu E_e, ktorá je jednoznačne určená práve obdržaným verejným kľúčom e. Dostane tak šifrový text $c = E_e(m)$, ktorý môže Bobovi poslať cez nezabezpečený kanál, pretože podľa vyššie uvedených predpokladov je k dešifrovaniu c nutná znalosť privátneho kľúča d. Vlastníkom tohoto kľúča je výhradne Bob, ktorý je pomocou neho schopný zo šifrového textu c rekonštruovať pôvodnú správu m tak, že použije inverznú transformáciu D_d a podľa definície obdrží:

$$m = D_d(E_e(m)) = D_d(c).$$

Najväčsou výhodou uvedeného spôsobu komunikácie je fakt, že v plnom rozsahu, a to vrátane distribúcie šifrovacieho (verejného) kľúča, prebieha cez nezabezpečený kanál (akékoľvek verejne dostupné médium, napríklad Internet). Táto výhoda je ale kompenzovaná nutnosťou riešenia celej rady problémov.

1.1.3. Motivácia

Šifrové texty by podľa svojho účelu, ktorým je zaistenie dôvernosti, mali v ideálnom prípade vykazovať rovnaké vlastnosti, ako bezpečná uzamykateľná schránka. Táto často uplatňovaná analógia vyzerá nasledovne – Alica vloží do schránky správu m a následne túto schránku zabezpečí nasadením kladky e, od ktorej vlastní kľúč d jedine Bob, a preto sa k uzamknutej správe nikto iný nedostane. Dosiahnutiu podobného stavu v praxi zabraňujú principiálne komplikácie determinované charakterom prenášaných informácií, ktorými je v konečnom dôsledku

bitový reťazec namiesto hmotnej správy uloženej v zamknutej schránke. Toto pozorovanie so sebou prináša niekoľko zásadných problémov:

- bitové reťazce môžu byť pri prenose po nezabezpečenom kanále sledované, čo znamená sprostredkovanie istej informácie o samotnej správe. Naproti tomu schránka zabezpečená kladkou neposkytuje pri svojom prenose pozorovateľovi žiadnu informáciu o správe, ktorá je uložená vo vnútri. Ak napríklad Alica pošle Bobovi dve správy v zabezpečených schránkach, nikto nie je schopný z prostého pozorovania týchto schránok posúdiť, či vo vnútri obsahujú zhodné správy alebo nie. Podobne by to malo fungovať v prípade šifrových systémov s verejným kľúčom. Naplnenie tejto požiadavky vylučuje praktické použitie šifrových systémov, ktoré vždy šifrujú rovnakú správu na rovnaký šifrový text (tzv. deterministické šifrové systémy).
- bitové reťazce môžu byť replikované, čo predstavuje možnosť realizácie tzv. opakovacích útokov. V prípade správ prenášaných v schránkach s kladkami je toto nerealizovateľné.
- bitové reťazce môžu byť modifikované, čím vznikajú iné šifrové texty. Prevrátením hodnoty ľubovoľného bitu z 0 na 1 alebo opačne vznikne nový šifrový text, čo zároveň mení aj zašifrovanú správu. Ak je navyše možné šifrový text upraviť na iný šifrový text tak, aby odpovedajúce správy zmysluplne súviseli, potom daný šifrový systém trpí slabinou, ktorá sa nazýva ohybnosť. Popísaná slabina nemá v prípade schránok s kladkami obdobu, pretože modifikácia schránky nijak neovplyvňuje správu uloženú vnútri.
- každý bitový reťazec je potenciálne šifrovým textom odpovedajúcim nejakej správe. Tento fakt môže byť zneužitý pri aktívnych útokoch na šifrový systém, kedy nepriateľ posiela ním vytvorené šifrové texty legitímnym účastníkom komunikácie. Napríklad môže Bobovi posielať ľubovoľné bitové reťazce, ktorý ich môže považovať za šifrové texty a preto sa bude snažiť dešifrovať ich. Pri tomto úsilí môže Bob reagovať rôznymi spôsobmi v závislosti na aktuálne dešifrovanom bitovom reťazci. Analýzou týchto reakcií môže útočník získať netriviálne informácie o danom šifrovom systéme vedúce až k jeho zlomeniu. Popísaný útok sa nazýva útok s voľbou šifrového textu. Predstava takého útoku v prípade schránok s kladkami je úsmevná.

Ukazuje sa, že neohybnosť šifrových systémov a ich odolnosť voči útokom s voľbou šifrového textu sú vzájomne ekvivalentné vlastnosti a ich splňovanie je zároveň nutnou a postačujúcou podmienkou pre dosiahnutie úrovne bezpečnosti, ktorá je intuitívne vnímaná ako dostatočná. Uvedenými problémami sa budeme podrobne zaoberať ďalej, no najskôr si ich dôsledky ilustrujeme na príklade šifrového systému RSA.

1.2. Útoky voči kryptosystémom s verejným kľúčom

V nasledujúcich oddieloch si predstavíme spôsob, akým sa modelujú útoky voči kryptosystémom. Konkrétne definujeme scenáre tých typov útokov, ktoré sú pre túto prácu relevantné. Detailnú klasifikáciu ďalších typov útokov voči kryptosystémom s verejným kľúčom vrátane vzájomných vzťahov medzi nimi je možné nájsť v práci [**1**].

1.2.1. Útok s voľbou otvoreného textu

Začneme popisom scenáru priebehu útoku s voľbou otvoreného textu:

Krok 1:
Prebehne generácia verejného a privátneho kľúča (e, d) požadovanej veľkosti. Útočník obdrží verejný kľúč, privátny nie.

Krok 2:
Útočník zadáva ľubovoľné dotazy šifrovaciemu orákulu. Každý dotaz je tvorený správou x, ktorú šifrovacie orákulum zašifruje. Výsledný šifrový text je vrátený útočníkovi.

Krok 3:
Útočník vytvorí ľubovoľné dve správy x_0, x_1 a pošle ich šifrovaciemu orákulu. To náhodne zvolí bit $b \in \{0, 1\}$ a následne zašifruje správu x_b. Výsledný cieľový šifrový text $y^* = E_e(x_b)$ odovzdá útočníkovi.

Krok 4:
Útočník pokračuje v kladení ľubovoľných dotazov x šifrovaciemu orákulu.

Krok 5:
Útočník vyprodukuje bit $b'_{\text{CPA}} \in \{0, 1\}$ reprezentujúci jeho tip hodnoty bitu b. V tomto kroku sa teda útočník na základe odpovedí šifrovacieho orákula a vlastných výpočtov rozhoduje, ktorej zo správ x_0, x_1 cieľový šifrový text y^* podľa neho odpovedá.

Poznámka. Uvedená špecifikácia popisuje iba jednu z možných variant útoku s voľbou otvoreného textu, ktorá je v odbornej literatúre známa ako "left or right CPA". Ďalšou, ekvivalentnou variantou je napríklad "real or random CPA", kde útočník v Kroku 3 pošle šifrovaciemu orákulu iba jednu správu x_0. Šifrovacie orákulum si druhú správu zvolí náhodne a obe správy pošle zašifrované útočníkovi. Ten sa následne snaží zistiť, ktorý šifrový text odpovedá správe x_0.

Definícia (bezpečnosť voči útoku s voľbou otvoreného textu). Výraz

$$\text{Adv}_{\mathcal{A}}^{\text{CPA}} = 2 \cdot \Pr[b'_{\text{CPA}} = b] - 1$$

vyjadruje *výhodu* útočníka $\mathcal{A}$ pri útoku s voľbou otvoreného textu. Kryptosystém sa nazýva *bezpečným voči útoku s voľbou otvoreného textu*, ak pre každého pravdepodobnostného polynomiálneho (vo veľkosti privátneho kľúča) nepriateľa $\mathcal{A}$ je výraz $\text{Adv}_{\mathcal{A}}^{\text{CPA}}$ zanedbateľnou funkciou vo veľkosti privátneho kľúča.

Teoreticky preukázaná bezpečnosť kryptosystému voči útoku s voľbou otvoreného textu je nutným, nie však postačujúcim, bezpečnostným požiadavkom pre praktické nasadenie daného kryptosystému.

1.2.2. Útok s voľbou šifrového textu

Nasleduje popis scenáru priebehu adaptívneho útoku s voľbou šifrového textu:

Krok 1:
Prebehne generácia verejného a privátneho kľúča (e, d) požadovanej veľkosti.

Útočník $\mathcal{A}$ obdrží verejný kľúč, privátny nie.

Krok 2:
Útočník zadáva ľubovoľné dotazy dešifrovaciemu orákulu. Každý dotaz je tvorený šifrovým textom y, ktorý dešifrovacie orákulum využitím znalosti privátneho kľúča dešifruje. Výsledná správa je vrátená útočníkovi. Šifrové texty y môže útočník vytvoriť ľubovoľným spôsobom (nie nutne korektnou aplikáciou príslušnej šifrovacej transformácie).

Krok 3:
Útočník vytvorí ľubovoľné dve správy x_0, x_1 a pošle ich šifrovaciemu orákulu. To náhodne zvolí bit $b \in \{0, 1\}$ a následne zašifruje správu x_b. Výsledný cieľový šifrový text $y^* = E_e(x_b)$ odovzdá útočníkovi.

Krok 4:
Útočník pokračuje v kladení ľubovoľných dotazov y dešifrovaciemu orákulu s jedinou obmedzujúcou podmienkou – žiaden dotaz nesmie obsahovať y^*.

Krok 5:
Útočník vyprodukuje bit $b'_{\mathrm{CCA2}} \in \{0, 1\}$ reprezentujúci jeho tip hodnoty bitu b. V tomto kroku sa teda útočník na základe odpovedí dešifrovacieho orákula a vlastných výpočtov rozhoduje, ktorej zo správ x_0, x_1 cieľový šifrový text y^* podľa neho odpovedá.

Poznámka. Uvedená špecifikácia opäť popisuje iba jednu z možných variant útoku s voľbou otvoreného textu, ktorá je v odbornej literatúre známa ako "left or right CCA2". Analogicky ako v podobnej poznámke v predchádzajúcom oddiele, aj tu existuje ekvivalentná varianta "real or random CCA2".

Definícia (bezpečnosť voči útoku s voľbou šifrového textu). Výraz

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{CCA2}} = 2 \cdot \Pr[b'_{CCA2} = b] - 1$$

vyjadruje *výhodu* útočníka $\mathcal{A}$ pri útoku s voľbou šifrového textu. Kryptosystém sa nazýva *bezpečným voči útoku s voľbou šifrového textu*, ak pre každého pravdepodobnostného polynomiálneho (vo veľkosti privátneho kľúča) útočníka $\mathcal{A}$ je výraz $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{CCA2}}$ zanedbateľnou funkciou vo veľkosti privátneho kľúča, teda ak platí $\Pr[b'_{CCA2} = b] \approx \frac{1}{2}$.

Teoreticky preukázaná bezpečnosť kryptosystému voči popísanému typu útoku, ktorý je najsilnejším bežne používaný modelom útoku (zahrnuje najviac možností, ako môže útok vyzerať), je vnímaná ako dostatočný argument pre bezpečnost daného kryptosystému pri praktickom nasadení. Avšak ako si ukážeme ďalej, teoretické dôkazy bezpečnosti často dosahujú iba asymptotické výsledky – typicky splňujú cieľovú úroveň bezpečnosti pre všetky dostatočne veľké hodnoty bezpečnostných parametrov. Túto problematiku sa snaží riešiť tzv. konkrétna bezpečnosť, ktorej predmetom je presné určenie praktických implikácií daného dôkazu, napríklad minimálnej veľkosti bezpečnostných parametrov daného kryptosystému v praxi pre zachovanie úrovne bezpečnosti garantovanej asymptotickým dôkazom.

Poznámka. Okrem adaptívneho útoku s voľbou šifrového textu (CCA2) existuje aj neadaptívna varianta (CCA1). Jej scenár priebehu sa od predchádzajúceho popisu líši iba absenciou Kroku 4.

1.3. Redukcionistická bezpečnosť

Redukcionistická bezpečnosť využíva princíp redukcie. Myšlienka tohoto prístupu spočíva v demonštrácii, že výpočtová nedosažiteľnosť problému $\mathcal{P}_1$ implikuje výpočtovú nedosažiteľnosť problému $\mathcal{P}_2$ (ekvivalentne výpočtová dosažiteľnosť problému $\mathcal{P}_2$ implikuje výpočtovú dosažiteľnosť problému $\mathcal{P}_1$) nasledujúcim spôsobom. Problém $\mathcal{P}_1$ je nejaký obtiažny matematický problém, ktorý je všeobecne považovaný za nedosažiteľný (napríklad problém faktorizácie celých čísel alebo problém diskrétneho logaritmu), a problém $\mathcal{P}_2$ predstavuje určitý typ útoku voči uvažovanému kryptosystému. Cieľom je ukázať, že každý, kto má k dispozícii efektívny algoritmus riešiaci problém $\mathcal{P}_2$, ho dokáže efektívnym spôsobom využiť k vyriešeniu problému $\mathcal{P}_1$. V tomto prípade hovoríme, že problém $\mathcal{P}_1$ sa redukuje na problém $\mathcal{P}_2$.

Pri dokazovaní bezpečnosti kryptosystému redukcionistickým prístupom sa typicky postupuje sporom. Na začiatku sa predpokladá, že existuje efektívny útočník schopný realizovať určitý typ útoku voči uvažovanému kryptosystému. K takému útočníkovi sa skonštruuje simulátor, ktorý útočníkovi simuluje odpovede napadnutého kryptosystému a tiež odpovede náhodného orákula. Tento simulátor následne využije útočníka ako svoj podprogram. Útočníkova schopnosť atakovať daný kryptosystém simulátoru umožní vyriešiť príslušný matematický problém, čím sa dosiahne spor s jeho predpokladanou výpočtovou nedosažiteľnosťou. Preto nemôže existovať útočník, ktorý dokáže úspešne (tj. s pravdepodobnosťou, ktorá nie je zanedbateľná) realizovať uvažovaný typ útoku voči danému kryptosystému.

1.3.1. Štandardný model

Štandardný model je klasický výpočtový model, v ktorom sa pri dokazovaní bezpečnosti kryptosystémov nevyužívajú tzv. ideálne prvky, akými sú napríklad náhodné orákulá. V dôkazoch bezpečnosti v štandardnom modele teda pracujeme väčšinou iba s efektívnymi algoritmami a nejakými výpočtovými predpokladmi (tj. že nejaký dobre známy problém, akým je napríklad vyššie spomínaná faktorizácia dostatočne veľkých prirodzených čísel, je pre efektívny algoritmus výpočtovo nedosažiteľné vyriešiť). Útočník je tu obmedzený jedine svojimi výpočtovými zdrojmi (obvykle je čas výpočtu polynomiálny v dĺžke vstupu). Preto typickým cieľom dôkazov bezpečnosti v štandardnom modele je ukázať, že kryptosystém je založený na výpočtovom predpoklade, ktorého využitie v konštrukcii daného kryptosystému garantuje neschopnosť útočníka efektívne vypočítať čokoľvek, čo by mohlo viesť k zlomeniu bezpečnosti tohto kryptosystému. Vďaka tomu sú dôkazy bezpečnosti v štandardnom modele notoricky známe svojou náročnosťou po myšlienkovej i technickej stránke.

1.3.2. Model náhodného orákula

Definícia. *Náhodným orákulom* $O : \{0,1\}^k \to \{0,1\}^l$ rozumieme verejne prístupnú entitu odpovedajúcu na dotazy $x \in \{0,1\}^k$ hodnotami $y \in \{0,1\}^l$. Odpovede O sú realizované jedinou funkciou $o : \{0,1\}^k \to \{0,1\}^l$, pričom platí:

- funkcia o je zvolená uniformne z množiny všetkých funkcií $\{0,1\}^k \to \{0,1\}^l$.
- popis funkcie o nie je nikomu (okrem samotného náhodného orákula O) známy.

Poznámka. Požiadavky predchádzajúcej definície znamenajú, že existuje iba jediný spôsob, ako je možné zistiť konkrétnu hodnotu $O(x) = o(x)$ pre ľubovoľné $x \in \{0,1\}^k$, a to priamo sa na ňu dotázať náhodného orákula O.

Náhodné orákulá sú ideálnym prvkom, pomocou ktorého sa v dôkazoch bezpečnosti modelujú hashovacie funkcie. Ideálnym prvkom, pretože žiadna reálna hashovacia funkcia z princípu nemôže vyhovovať uvedenej definícii (popis hashovacej funkcie je v realite dopredu známy a obvykle je realizovateľný malým počtom krokov). Tento model však do istej miery aproximuje dôležité vlastnosti hashovacích funkcií, akými sú:

- náhodnosť – funkcia o v predchádzajúcej definícii je volená náhodne;
- jednosmernosť – invertovať funkčnú hodnotu je výpočtovo nedosažiteľná úloha, pretože popis funkcie o je známy výhradne náhodnému orákulu;
- konzistencia – na rovnaký dotaz odpovedá náhodné orákulum rovnakou odpoveďou, pretože o je funkcia.

Ťažisko metodiky náhodného orákula spočíva v spôsobe, akým simulátor simuluje útočníkovi náhodné orákulum. Simulátor je totiž obmedzovaný iba dodržaním troch podmienok uvedených vyššie – tie v samotnom dôkaze zaručujú potrebnú nerozlíšiteľnosť medzi reálnym útokom a simuláciou z pohľadu útočníka. Trik je v tom, že simulátor niektoré odpovede na dotazy útočníka definuje (falzifikuje) tak, aby využil útočníkovu schopnosť zlomiť daný kryptosystém vo svoj prospech. Toto môže urobiť ľubovoľným spôsobom – platiť musí len to, aby jeho odpovede útočníkovi splňovali podmienky náhodnosti, jednosmernosti a konzistencie. Táto vlastnosť metodiky náhodného orákula je jej kritikmi často označovaná ako sporná (viď napríklad [**13**]).

Simulácia hashovacej funkcie pomocou náhodného orákula prebieha nasledovne. Na začiatku sa pripraví prázdny zoznam, do ktorého sa v priebehu útoku zapisujú jednotlivé dvojice dotaz-odpoveď. Tento zoznam je verejne prístupný s prístupom len pre čítanie, aby bola zaistená konzistencia. Pri dotaze na nejakú hodnotu simulovanej hashovacej fukcie sa najprv overí, či k danému dotazu už neexistuje záznam v zozname dvojíc dotaz-odpoveď. Ak áno, je vrátená odpovedajúca hodnota. V opačnom prípade sa na základe odpovede náhodného orákula najskôr pridá do zoznamu nová dvojica a následne sa odpovie dotazovateľovi.

V nasledujúcich kapitolách sa s metodikou náhodného orákula zoznámime podrobnejšie na konkrétnych prípadoch z praxe. Najskôr si ale ukážeme historicky prvé a veľmi názorné využitie metodiky náhodného orákula pre dôkaz bezpečnosti Rabinovho kryptosystému.

1.4. Rabinov kryptosystém

V tomto oddiele budeme čerpať z článkov [**17**] a [**13**].

1.4.1. Popis

Generácia kľúča:

1) zvolia sa dve rôzne dostatočne veľké prvočísla p, q a spočíta sa ich súčin $n = pq$, ktorý ďalej slúži ako modul – Rabinov kryptosystém totiž pracuje v okruhu $\mathbb{Z}_n$. Dostatočne veľké prvočíslo má v súčasnosti veľkosť rádovo 1024 bitov, takže n bude veľké rádovo 2048 bitov. Prvočíselnosť takto veľkých čísel sa overuje pomocou efektívnych pravdepodobnostných testov.
2) verejný kľúč je hodnota n.
3) privátny kľúč tvorí dvojica (p, q).

Šifrovanie a dešifrovanie:

1) Bob vygeneruje kľúč a následne distribuuje jeho verejnú časť n Alici.
2) Alica chce Bobovi poslať správu m spĺňajúcu $1 \leq m < n$. Vytvorí teda šifrový text $c \equiv m^2 \mod n$, ktorý odošle Bobovi. Šifrový text Rabinovho kryptosystému je teda vždy kvadratickým reziduom modulo n.
3) Keďže šifrovacia transformácia Rabinovho kryptosystému nie je injektívne zobrazenie (v skutočnosti sa jedná o zobrazenie 4:1), dešifrovanie nie je jednoznačné. Bob teda obdrží šifrový text c a spočíta:

$$m_p = \sqrt{c} \mod p,$$

$$m_q = \sqrt{c} \mod q.$$

V prípade, že platí $p \equiv q \equiv 3 \mod 4$ (generácia kľúča často prebieha podľa tejto podmienky), spočíta Bob príslušné hodnoty využitím Legendreových symbolov:

$$m_p = c^{\frac{p+1}{4}} \mod p,$$

$$m_q = c^{\frac{q+1}{4}} \mod q.$$

Platnosť uvedenej kongruencie nie je pre fungovanie Rabinovho kryptosystému nutnosťou, avšak v opačnom prípade je výpočet hodnôt m_p, m_q komplikovanejší (napr. pomocou Berlekampovho algoritmu). V nasledujúcom kroku spočíta Bob využitím rozšíreného Euklidovho algoritmu hodnoty y_p, y_q tak, aby platilo:

$$p \cdot y_p + q \cdot y_q = 1.$$

Nakoniec, pomocou Čínskej vety o zbytkoch, určí Bob všetkých potenciálnych kandidátov na správu m ako:

$$r = (y_p \cdot p \cdot m_q + y_q \cdot q \cdot m_p) \mod n,$$

$$-r = n - r,$$

$$s = (y_p \cdot p \cdot m_q - y_q \cdot q \cdot m_p) \mod n,$$

$$-s = n - s.$$

Z týchto štyroch hodnôt si Bob vyberie ako správu m tú, ktorá mu dáva najväčší zmysel.

1.4.2. Bezpečnosť

TVRDENIE 1.1. *Označme* $c \equiv m^2 \mod n$ *ako šifrový text Rabinovho kryptosystému odpovedajúci správe* m*. Potom neexistuje pravdepodobnostný polynomiálny (vo veľkosti privátneho kľúča) algoritmus pre rekonštrukciu správy* m*, ktorý ako vstup obdrží šifrový text* c *a verejný kľúč* n*.*

DÔKAZ. Pre spor predpokladajme, že existuje útočník – pravdepodobnostný polynomiálny (vo veľkosti privátneho kľúča) algoritmus – ktorého vstupom sú hodnoty c, n. Nech tento útočník dokáže s pravdepodobnosťou, ktorá nie je zanedbateľná, z uvedených vstupov vypočítať odpovedajúcu správu m. Ukážeme, ako môže simulátor – pravdepodobnostný polynomiálny (vo veľkosti privátneho kľúča) algoritmus – ktorý využije útočníka ako svoj podprogram, faktorizovať modul n (opäť s pravdepodobnosťou, ktorá nie je zanedbateľná). Simulátor začne tým, že zvolí náhodne nejakú hodnotu x, $1 \leq x < n$. Spočíta $c \equiv x^2 \mod n$ a túto hodnotu predá útočníkovi spoločne s modulom n ako vstup. Útočník vráti simulátoru ako svoj výstup správu m. Táto správa je podľa predchádzajúceho oddielu jednou zo štvorice hodnôt r, $-r$, s, $-s$. Nech teda napríklad platí:

$$x = r = (y_p \cdot p \cdot m_q + y_q \cdot q \cdot m_p) \mod n.$$

S pravdepodobnosťou $1/2$ vráti útočník simulátoru ako správu m jednu z hodnôt $r = x$ alebo $-r = -x$. V tomto prípade simulátor z výstupu útočníka nemá žiadny úžitok, preto zvolí novú hodnotu x a celý proces opakuje. Pravdepodobnosť, že po k opakovaniach simulátor aspoň raz neobdrží od útočníka správu m rovnú jednej z hodnôt s alebo $-s$, je rovná $1/2^k$, takže po dostatočnom počte opakovaní sa simulátor nakoniec dočká. Predpokladajme teda, že správa obdržaná simulátorom od útočníka splňuje:

$$m = s = (y_p \cdot p \cdot m_q - y_q \cdot q \cdot m_p) \mod n.$$

Simulátor následne spočíta:

$$y = x - m = r - s = 2y_q \cdot q \cdot m_p \mod n,$$

pričom y je nenulový prvok okruhu $\mathbb{Z}_n$, ktorý má netriviálneho najväčšieho spoločného deliteľa s modulom n, pretože jedným z jeho faktorov je q. Simulátor preto využije rozšírený Euklidov algoritmus, aby spočítal:

$$gcd(y, n) = q \mod n,$$

čím sa mu úspešne podarilo faktorizovať modul n. To ale predstavuje spor s výpočtovou nedosažiteľnosťou problému faktorizácie celých čísel, takže útočník popísaný v tomto dôkaze nemôže existovať. V prípade, kedy simulátor od útočníka obdrží ako správu $m = -s$, sa postupuje analogicky.

□

Uvedený dôkaz žiaľ platí len pre prípad útoku s voľbou otvoreného textu a negarantuje teda dostatočnú úroveň bezpečnosti pre praktické použitie. Ukážeme si, prečo. Uvažujme útočníka, ktorému povolíme prístup k dešifrovaciemu orákulu (prípad útoku s voľbou šifrového textu). Taký útočník môže kompletne kompromitovať bezpečnosť kryptosystému – stačí, ak bude postupovať rovnakým spôsobom, ako simulátor v uvedenom dôkaze, čím jednoducho faktorizuje modul n. Takto získa privátny kľúč (p, q) a môže dešifrovať ľubovoľný šifrový text.

Predchádzajúci odstavec je dôvodom, prečo sa Rabinov kryptosystém neujal v praxi. Zaujímavou je však varianta Rabinovho kryptosystému, ktorú publikoval Boneh v článku [**5**] a ktorá je známa ako Boneh-Rabinov kryptosystém. Táto varianta využíva šifrovaciu transformáciu pôvodného Rabinovho kryptosystému (umocňovanie na druhú modulo n), ale pred jej aplikáciou najskôr kóduje správu m podľa modifikovaného protokolu OAEP (viď nasledujúca kapitola). Boneh-Rabinov kryptosystém je preukázateľne bezpečný voči adaptívnemu útoku s voľbou šifrového textu a príslušný dôkaz bezpečnosti dokonca netrpí ani problémom nedostatočne priliehavej redukcie, ako je tomu v prípade RSA-OAEP (viď oddiel 3.1).

2

Bezpečnosť RSA-OAEP a model náhodného orákula

2.1. Kryptosystém RSA

Účelom tohto oddielu je charakterizácia pôvodného kryptosystému RSA navrhnutého Rivestom, Shamirom a Adlemanom v práci [**18**], vrátane demonštrácie problémov asymetrických kryptosystémov z predchádzajúcej kapitoly. Oddiel uzatvára stručný popis jedného z prvých štandardizovaných riešení týchto problémov v praxi.

2.1.1. Popis

Generácia kľúča:

1) zvolia sa dve rôzne dostatočne veľké prvočísla p, q a spočíta sa ich súčin $n = pq$, ktorý ďalej slúži ako modul – kryptosystém RSA totiž pracuje v okruhu $\mathbb{Z}_n$.
2) spočíta sa funkčná hodnota Eulerovej funkcie v bode n, teda číslo $\phi(n) = (p-1)(q-1)$.
3) zvolí sa číslo e vyhovujúce nerovnosti $1 < e < \phi(n)$ a podmienke $\gcd(e, \phi(n)) = 1$. Bežná voľba je $e = 65537$. Verejný kľúč tvorí dvojica (n, e).
4) použitím rozšíreného Euklidovho algoritmu sa dopočíta číslo d splňujúce kongruenciu $ed \equiv 1 \mod \phi(n)$. Hodnota $\phi(n)$ sa zničí a privátny kľúč tvorený dvojicou (n, d) sa uchová v tajnosti.

Poznámka. Dostatočná veľkosť modulu n eliminuje možnosť kompromitácie kryptosystému RSA hrubou silou – faktorizáciou modulu n. Rekonštrukciou p, q totiž útočník získa aj hodnotu $(p-1)(q-1) = \phi(n)$, pomocou ktorej už triviálne dopočíta d podľa bodu 4) z predchádzajúceho odstavca.

Šifrovanie a dešifrovanie:

1) Bob vygeneruje kľúč a následne distribuuje jeho verejnú časť (n, e) Alici.
2) Alica chce Bobovi poslať správu m splňujúcu $1 \leq m < n$. Vytvorí teda šifrový text $c \equiv m^e \mod n$, ktorý odošle Bobovi.
3) Bob obdrží šifrový text c a spočíta $c^d \equiv m^{ed} \equiv m^{\phi(n)+1} \equiv m^{\phi(n)} m \equiv m \mod n$ podľa Eulerovej vety.

2.1.2. Základné problémy

Klasický kryptosystém RSA je v popísanej podobe prakticky nepoužiteľný. Trpí totiž hneď niekoľkými zásadnými bezpečnostnými slabinami, ktoré ho veľmi vzdaľujú od predstavy bezpečnej schránky s kladkou slúžiacej k prenosu informácií

vyžadujúcich utajenie. Tieto slabiny si teraz predstavíme.

Kryptosystém RSA je deterministický:
Dve zhodné správy teda RSA zašifruje vždy na zhodný šifrový text. Toto predstavuje problém hlavne v prípade, keď je množina možných správ relatívne malá. Vtedy je nepriateľ schopný vypočítať ku všetkým správam odpovedajúce hodnoty šifrových textov. Prebiehajúcu komunikáciu tak len pasívne sleduje a porovnáva s vypočítanými hodnotami možných šifrových textov.

Kryptosystém RSA je ohybný:
Ak ľubovoľný šifrový text $c \equiv m^e \mod n$ nepriateľ modifikuje prenásobením nejakou hodnotou x^e, obdrží šifrový text $c_0 \equiv m^e x^e \equiv (mx)^e \mod n$. Vzájomná korelácia príslušných zašifrovaných správ m a $mx \mod n$ je teda prostredníctvom vhodnej voľby x nastaviteľná podľa vôle. Ako príklad možnosti zneužitia tejto vlastnosti dobre poslúži nasledujúca situácia.

Bob vypísal výberové konanie na veľkú zakázku a prijíma ponuky. Rozhoduje sa podľa najnižšej ceny. Alica má o túto zákazku záujem. Svoju konkrétnu cenovú ponuku pošle Bobovi v zašifrovanom tvare, aby ju neprekonala Eva z konkurenčnej spoločnosti. Ak je však Eva schopná odchytiť zašifrovanú správu obsahujúcu cenovú ponuku od Alice, dokáže Bobovi ľahko ponúknuť nižšiu cenu než Alica. A to bez toho, aby vôbec poznala jej konkrétnu hodnotu – stačí, aby vhodne zvolila číslo x (napr. $x \equiv \frac{9}{10} \mod n$) a šifrový text od Alice prenásobila vyššie uvedeným spôsobom.

Kryptosystém RSA je zraniteľný útokom s voľbou šifrového textu:
Cieľom tohoto útoku je rekonštrukcia správy m zo šifrového textu $c = m^e \mod n$ bez znalosti hodnoty privátneho kľúča d. Útočník zvolí ľubovoľne hodnotu y a vytvorí nový šifrový text $c_0 = y^e c = (ym)^e$. Ak sa útočníkovi podarí dosiahnuť, aby mu vlastník privátneho kľúča tento zdanlivo neškodný šifrový text c_0 dešifroval, potom ako výsledok obdrží hodnotu $ym \mod n$, odkiaľ už pomocou znalosti y ľahko určí hodnotu pôvodnej správy m.

2.1.3. Doplnenie správy o náhodnú hodnotu

Kryptosystém RSA sa nikdy v praxi nepoužíval vo svojej pôvodnej podobe. Práve kvôli existencii uvedených problémov bol za účelom ich eliminácie rôzne vylepšovaný. Dlho používaným riešením bolo zreťazenie správy m pred jej zašifrovaním s náhodne vygenerovanou nenulovou hodnotou r, ktorá mala pevne stanovenú dĺžku (aspoň 8 bajtov). Šifrovanie teda prebiehalo stále rovnako, akurát sa miesto správy m šifrovalo jednoduché zreťazenie $p(m, r) = (00||02||r||00||m)$, kde symboly 00, resp. 02 označujú bajty príslušnej hodnoty (v hexadecimálnej sústave). Pri dešifrovaní sa najskôr overilo, že v obdržanom $p(m, r)$ majú prvé dva bajty hodnotu $(00||02)$ a že ďalej po aspoň ôsmych nenulových bajtoch nasleduje aspoň jeden nulový bajt. Nakoniec sa oddelili už nepotrebné časti (všetky pevne určené bajty a náhodný kus r), čo bolo možné vďaka ich predom stanovenej hodnote a pozícii. Účelom takého náhodného dopĺňania bolo zavedenie nedeterminizmu do kryptosystému RSA – zhodné správy boli pred šifrovaním vždy doplnené o rôzne náhodné hodnoty a tak sa už viac nešifrovali na zhodné šifrové texty.

Toto riešenie sa používalo ako štandardizované (PKCS #1 do verzie 1.5) v protokoloch SSL/TLS až do roku 1998, kedy Bleichenbhacher v práci [**3**] publikoval prakticky použiteľný adaptívny útok s voľbou šifrového textu, pomocou ktorého je možné lúštiť zašifrovanú správu. Útočník má pri tomto útoku k dispozícii nejaký šifrový text $c = (p(m, r))^e \mod n$ a chce z neho získať hodnotu správy m. Vlastníkovi privátneho kľúča (dešifrovaciemu orákulu) za týmto účelom odošle mnoho šifrových textov tvaru $c' = x^e c \mod n$ pre náhodne volené hodnoty $x \in \mathbb{Z}_n$. Ten sa pri obdržaní šifrového textu pokúsi o dešifrovanie výpočtom $a = (c')^d \mod n$ a následným určením m', pre ktoré je $a = p(m', r)$ pre nejaké náhodné r. Ak bude hodnota a vyhovovať predpísanému formátu, tak sa mu to podarí, inak nie. Všetko, čo je potrebné, aby útok fungoval, je práve tento chybový kód zachycujúci výsledok jednotlivých pokusov o dešifrovanie c'. Bleichenbacher vo svojej práci ukazuje, ako špeciálny program, ktorý na vstupe dostane spomenutý chybový kód, dokáže úspešne dešifrovať c a rekonštruovať tak pôvodnú správu m.

Bleichenbacherov útok však nebol jediným fungujúcim útokom na implementácie RSA využívajúce náhodné doplňanie správy pred zašifrovaním. Tak sa postupne objavila potreba podobné útoky nejakým spôsobom metodicky vylúčiť už pri samotnom návrhu implementácie kryptosystému, namiesto neustáleho ad-hoc zaisťovania bezpečnosti pri každom ďalšom objavení nepredvídaného útoku.

2.2. Optimal Asymmetric Encryption Padding

2.2.1. Popis OAEP

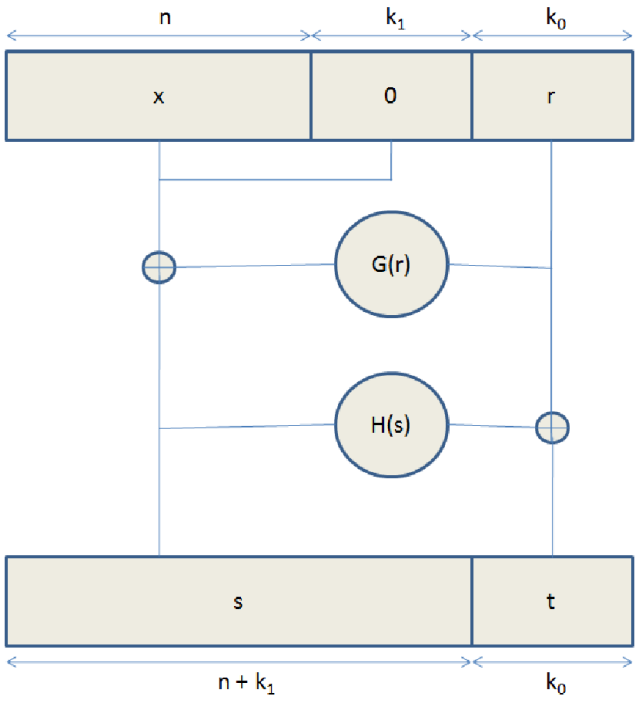
V roku 1994 navrhli Bellare a Rogaway v článku [**2**] protokol OAEP určený primárne pre šifrovanie správ pomocou RSA. Narozdiel od všetkých predchádzajúcich prakticky používaných protokolov pre kryptosystémy s verejným kľúčom, autori OAEP ako prví priniesli so samotnou konštrukciou aj akýsi dôkaz jej bezpečnosti. Cieľom bola teoretická eliminácia známych problémov RSA, ktorá by v praxi vylučovala úspešnú realizáciu útoku s voľbou šifrového textu. Predvedieme si ich prístup.

Nech f je jednosmerná permutácia s padacími vrátkami, teda bijektívne zobrazenie $f : \{0,1\}^k \to \{0,1\}^k$ a označme g inverzné zobrazenie príslušné k f. Ďalej uvažujme prirodzené čísla k_0, k_1 splňujúce $k_0 + k_1 = k$. Priestor možných správ pozostáva z prvkov $x \in \{0,1\}^n$, kde $n = k - (k_0 + k_1)$. Nakoniec nech $G : \{0,1\}^{k_0} \to \{0,1\}^{n+k_1}$ a $H : \{0,1\}^{n+k_1} \to \{0,1\}^{k_0}$ sú náhodné orákulá. Protokol OAEP funguje nasledovne (grafické znázornenie viď Obrázok 2.1):

Generácia kľúča: Vygeneruje sa jednosmerná permutácia f spoločne s padacími vrátkami (inverznou permutáciou) g. Šifrovacou permutáciou je teda f a dešifrovacou permutáciou je g.

Šifrovanie:

1) náhodne sa zvolí hodnota $r \in \{0,1\}^{k_0}$.
2) vypočíta sa hodnota $s = G(r) \oplus (x||0^{k_1}), s \in \{0,1\}^{n+k_1}$.
3) vypočíta sa hodnota $t = H(s) \oplus r, t \in \{0,1\}^{k_0}$.
4) určí sa hodnota $(s||t), (s||t) \in \{0,1\}^k$.



Obr. 2.1. Schéma OAEP

5) vypočíta sa hodnota $y = f(s||t), y \in \{0,1\}^k$.

Dešifrovanie:

1) vypočíta sa hodnota $(s||t) = g(y), (s||t) \in \{0,1\}^k$.
2) určí sa hodnota $s = (s||t)[n + k_1 + k_0 - 1 \ldots k_0], s \in \{0,1\}^{n+k_1}$.
3) určí sa hodnota $t = (s||t)[k_0 - 1 \ldots 0], t \in \{0,1\}^{k_0}$.
4) vypočíta sa hodnota $r = H(s) \oplus t, r \in \{0,1\}^{k_0}$.
5) vypočíta sa hodnota $z = G(r) \oplus s, z \in \{0,1\}^{n+k_1}$.
6) určí sa hodnota $x = z[n + k_1 - 1 \ldots k_1], x \in \{0,1\}^n$.
7) určí sa hodnota $c = z[k_1 - 1 \ldots 0], c \in \{0,1\}^{k_1}$.
8) ak $c = 0^{k_1}$, potom je výstupom správa x, inak je šifrový text y odmietnutý a výstupom je hláška `zlyhanie`.

Poznámka. Schéma OAEP predstavuje všeobecný protokol pre šifrovanie správ. V prípade, že funkcie f, g predstavujú po rade šifrovaciu a dešifrovaciu transformáciu RSA (teda $f(x) = x^e \mod n$ a $g(y) = y^d \mod n$), hovoríme o kryptosystéme RSA-OAEP.

Ozrejmime si trochu fungovanie popísaného protokolu. Doplnenie správy x veľkosti n o k_1 nulových bitov je potrebné, aby len malá časť všetkých reťazcov dĺžky k, konkrétne $\frac{2^{k-k_1}}{2^k} = \frac{1}{2^{k_1}}$, mohla byť prípustnou správou. Práve nastavením hodnoty parametra k_1 sa tento rozsah dá primerane regulovať. Ďalších k_0 náhodných bitov, ktoré sú pred zašifrovaním pridávané k správe x, slúžia k odstráneniu determinizmu šifrovacej schémy. Dôležité je tiež, že prístup k náhodným orákulám G, H je verejný, takže ktokoľvek môže získať príslušné funkčné hodnoty $G(r), H(s)$. V praxi sú totiž náhodné orákulá G, H realizované použitím hashovacích funkcií.

2.2.2. Pôvodný dôkaz bezpečnosti RSA-OAEP

TVRDENIE 2.1. *Šifrový systém RSA-OAEP je bezpečný voči útoku s voľbou šifrového textu.*

DÔKAZ. Pre spor predpokladajme, že existuje pravdepodobnostný polynomiálny (vo veľkosti privátneho kľúča d) útočník $\mathcal{A}$, pre ktorého je výraz $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{CCA2}}$ funkciou, ktorá nie je zanedbateľná. Našim cieľom je ukázať, že takýto útočník $\mathcal{A}$ môže byť použitý ako podprogram simulátora $\mathcal{S}$, ktorý na základe výstupov $\mathcal{A}$ dokáže k zadanému y nájsť hodnoty s, r tak, aby platilo $y \equiv (s||(r \oplus H(s)))^e \mod n$. Tento problém je vlastne problémom invertovania RSA (ak označíme $y \equiv x^e \mod n$, hľadáme $x = s||(r \oplus H(s))$ splňujúce predchádzajúcu rovnosť) a pre dostatočne veľký modul n sa v súčasnej dobe nevie efektívne (pravdepodobnostným polynomiálnym algoritmom) riešiť. Tento fakt využijeme v závere dôkazu k dosiahnutiu sporu.

Simulátor $\mathcal{S}$ bude pri simulovanom útoku s voľbou šifrového textu pre útočníka $\mathcal{A}$ zohrávať rolu dešifrovacieho orákula a tiež bude simulovať náhodné orákulá G, H. Simulátor $\mathcal{S}$ teda dostane na vstupe šifrový text y, ku ktorému chce nájsť e-tu odmocninu $\mod n$. Pri simulovanom útoku s voľbou šifrového textu postupuje nasledovne:

1) Pri simulácii náhodných orákul G, H simulátor $\mathcal{S}$ vracia náhodné hodnoty, ktoré si pamätá. To znamená, že útočníkovi na dotazy r, s, odpovedá náhodne vygenerovanými hodnotami $G(r), H(s)$, ktoré útočník potrebuje pre korektné šifrovanie správ. Príslušné dvojice $(r, G(r)), (s, H(s))$ si ukladá do dvoch samostatných zoznamov. Pred každou ďalšou odpoveďou na dotaz útočníka vždy najskôr prejde príslušný zoznam a v prípade, že k tomuto dotazu už vo svojom zozname má uloženú nejakú hodnotu, vráti túto uloženú hodnotu. Tým je zachovaná konzistencia jeho odpovedí v prípade opakovaných dotazov útočníka – na rovnaký dotaz vracia rovnakú odpoveď.
2) Pri simulácii dešifrovacieho orákula obdrží $\mathcal{S}$ od útočníka hodnotu y'. Prejde všetky uložené dvojice $(r, G(r)), (s, H(s))$ a nájde také, aby platilo $(s||(r \oplus H(s)))^e \equiv y' \mod n$. V prípade, že vyhovujúce dvojice nenájde, odpovie hláškou `zlyhanie`, pretože útočník sa snaží podvádzať – nemôže sa dotazovať na platný šifrový text y' bez toho, aby najskôr nebol vytvorený. Pritom nutným predpokladom pre vytvorenie ľubovoľného šifrového textu je podľa protokolu OAEP útočníkova znalosť príslušných hodnôt $G(r'), H(s')$, ktoré môže získať jedine dotazom na $\mathcal{S}$. Na druhej strane, ak simulátor $\mathcal{S}$ príslušné dvojice $(r', G(r')), (s', H(s'))$ vo svojich zoznamoch nájde, položí $(x^0)' = s' \oplus G(r')$ a overí, že posledných k_1 bitov $(x^0)'$ tvoria samé nuly. Ak netvoria, $\mathcal{S}$ odpovie hláškou `zlyhanie`, pretože sa jedná o neplatný šifrový text. V opačnom prípade útočníkovi ako dešifrovanú správu vráti prvých n bitov $(x^0)'$, teda hodnotu $x' = (x^0)'[n + k_1 - 1, \ldots, k_1]$.
3) Simulátor $\mathcal{S}$ obdrží od útočníka $\mathcal{A}$ dve správy x_0, x_1. Namiesto toho, aby náhodne zvolil bit $b \in \{0, 1\}$ a útočníkovi vrátil príslušný cieľový šifrový text $y^* = y_b$, však $\mathcal{S}$ položí $y^* = y$, kde y je šifrový text, ktorý sa $\mathcal{S}$ snaží dešifrovať. Tento krok ale neovplyvní útočníkovu schopnosť následne určiť s pravdepodobnosťou $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{CCA2}}$ hodnotu bitu $b'_{\mathrm{CCA2}} = b$ tak, akoby cieľový šifrový text y^* bol platným šifrovým textom odpovedajúcim správe x_b. A

to aj napriek tomu, že cieľový šifrový text y^* vznikol v skutočnosti inak a príslušné hodnoty $r_b, G(r_b), s_b, H(s_b)$ ešte vtedy v aktuálnych zoznamoch $\mathcal{S}$ ani nemusia existovať. Tento trik vychádza z vlastností náhodných orákul a platnosti nasledujúceho vzťahu:

$$(\forall y^*, x_b)(\exists r^*, G(r^*), s^*, H(s^*)) :$$

$$(s^*||(r^* \oplus H(s^*)))^e \equiv y^* \mod n) \ \& \ (x_b^0 = s^* \oplus G(r^*)).$$

Hodnota $G(r^*)$ je totiž náhodná a preto jediný spôsob, akým útočník dokáže získať netriviálnu informáciu o odpovedajúcej hodnote $x_b^0 = s^* \oplus G(r^*)$ a tým aj o hodnote x_b, je poznať hodnotu $G(r^*)$. Lenže na tú sa útočník najskôr musí dotázať simulátora $\mathcal{S}$, teda v priebehu svojho výpočtu musí simulátoru položiť dotaz s hodnotou $r' = r^*$, aby tak obdržal $G(r^*)$. Ale pretože $r^* = t^* \oplus H(s^*)$, kde hodnota $H(s^*)$ je taktiež náhodná, útočník musí v priebehu svojho výpočtu položiť simulátoru aj dotaz s hodnotou $s' = s^*$, aby tak obdržal $H(s^*)$.

4) Simulátor $\mathcal{S}$ vlastní zoznamy všetkých dotazov útočníka $\mathcal{A}$ aj s odpoveďami na tieto dotazy, teda zoznamy obsahujúce všetky hodnoty $(r, G(r)), (s, H(s))$ použité v priebehu výpočtu $\mathcal{A}$. Útočník dokáže správu x_b určiť správne s pravdepodobnosťou rovnou výrazu $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{CCA2}}$. Tento výraz je podľa predpokladu funkciou veľkosti privátneho kľúča, ktorá nie je zanedbateľná. Simulátor teda po obdržaní hodnoty $x_{b_{\mathrm{CCA2}}}$, od útočníka jednoducho prejde všetky dvojice (r, s) uložené vo svojich zoznamoch a pre každú z nich skontroluje, či platí rovnosť $(s||(r \oplus H(s)))^e \equiv y \mod n$. V momente, keď $\mathcal{S}$ narazí na vyhovujúcu dvojicu (r^*, s^*), úspešne vyriešil zadaný problém. To nastane s pravdepodobnosťou rovnou $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{CCA2}}$, ktorá nie je zanedbateľná. Nakoniec, útočník $\mathcal{A}$ je pravdepodobnostný polynomiálny algoritmus a teda môže vzniesť len polynomiálne mnoho dotazov. Simulátor $\mathcal{S}$ potrebuje v najhoršom prípade vyskúšať všetky možné dvojice (r, s). Časová zložitosť nájdenia dvojice (r^*, s^*) je preto $\mathcal{O}((\max\{\#r, \#s\})^2)$, kde $\#r, \#s$ označuje počet dotazov $\mathcal{A}$ na $\mathcal{S}$ s hodnotami r, s. Táto zložitosť je však stále polynomiálna, takže nájdenie (r^*, s^*) a tým aj úspešné invertovanie šifrového textu y je pre $\mathcal{S}$ výpočtovo dosažiteľná úloha. To predstavuje hľadaný spor.

□

2.2.3. Problémy pôvodného dôkazu bezpečnosti RSA-OAEP

Predchádzajúci dôkaz využíva vlastnosti náhodných orákul, ktoré povoľujú zaujímavý trik – hodnoty $r^*, G(r^*), s^*, H(s^*)$ odpovedajúce šifrovému textu $y^* = y$, môže $\mathcal{S}$ definovať až dodatočne. Podobná záležitosť nie je v reálnom svete možná, pretože každá hashovacia funkcia, ktorá sa v skutočnosti používa na mieste G, H, má pevne stanovený postup výpočtu funkčnej hodnoty (nechová sa ako náhodné orákulum). Tento rozpor predstavuje najväčší problém modelovania skutočných kryptosystémov a dokazovania ich bezpečnosti pomocou ideálnych prvkov, akými sú náhodné orákulá.

Okrem toho však uvedený dôkaz trpí ešte zásadnejším problémom, ktorý objavil Shoup a následne publikoval v práci [**19**]. Všimol si, že pri splnení istého predpokladu môže útočník využiť znalosti y^* k tomu, aby sa pri jeho dešifrovaní na hodnotu r^* vôbec nemusel simulátora dotázať, čo spochybňuje použitú

argumentáciu (konkrétne bod 3) vyššie uvedeného dôkazu). Tým predpokladom je existencia tzv. XOR-ohybnej jednosmernej permutácie s padacími vrátkami. Ukážeme si v stručnosti, ako to celé funguje.

Nech teda f_0 je XOR-ohybná jednosmerná permutácia s padacími vrátkami. To znamená, že zo znalosti dvojice hodnôt $f_0(x), a$ je útočník schopný odvodiť hodnotu $f_0(x \oplus a)$ s pravdepodobnosťou, ktorá nie je zanedbateľná. Definujme šifrovaciu transformáciu OAEP ako $f(s||t) = s||f_0(t)$. Vďaka vlastnostiam f_0 je f tiež jednosmernou permutáciou s padacími vrátkami. Označme $u = f_0(t)$. Potom pre cieľový šifrový text platia nasledujúce vzťahy:

$$y^* = f(s^*||t^*) = s^*||f_0(t^*) = s^*||u^*,$$
$$s^* = G(r^*) \oplus (m^*||0^{k_1}),$$
$$t^* = H(s^*) \oplus r^*.$$

Funkcia f je teda identitou na ľavých $(n + k_1)$ bitoch (hodnota s^* sa zobrazí sama na seba). Pre ľubovoľnú náhodnú hodnotu $\delta' \in \{0,1\}^n$ definujme výrazy $\Delta' = \delta'||0^{k_1}$ a $s' = s^* \oplus \Delta'$. Ďalej položme $t' = H(s') \oplus r^* = t^* \oplus (H(s^*) \oplus H(s'))$. Vďaka XOR-ohybnosti f_0 vieme zo znalosti $u = f_0(t^*)$ a výrazu $H(s^*) \oplus H(s')$ vypočítať hodnotu $u' = f_0(t')$ s pravdepodobnosťou, ktorá nie je zanedbateľná. Nakoniec, výraz $y' = s'||u'$ je platným šifrovým textom odpovedajúcim správe $m' = m^* \oplus \delta'$, pričom $r' = r^*$, pretože

$$t' = f_0^{-1}(u') = t^* \oplus (H(s^*) \oplus H(s')) = H(s') \oplus r^*,$$
$$r' = H(s') \oplus t' = r^*,$$
$$s' \oplus G(r') = \Delta' \oplus s^* \oplus G(r^*) = \Delta' \oplus (m^*||0^{k_1}) = (m^* \oplus \delta')||0^{k_1}.$$

Útočník si teda z obdržaného cieľového textu y^* odvodí nový platný šifrový text y' a následne sa dešifrovacieho orákula dotáže na správu m', odpovedajúcu tomuto šifrovému textu. Pomocou znalosti hodnôt m', δ' už ľahko dopočíta cieľovú správu m^*, lebo ako sme si pred chvíľou ukázali, platí $m^* = m' \oplus \delta'$.

Keďže pri tomto postupe je útočníkov program v medziach scenára adaptívneho útoku s voľbou šifrového textu, dostávame protipríklad k dôkazu bezpečnosti OAEP voči tomuto útoku. Dôvodom je, že útočník dokáže platný šifrový text y' zostrojiť bez toho, aby sa náhodného orákula G dotázal na odpovedajúcu hodnotu $r' = r^*$. Namiesto tohto dotazu len využije platnosť vyššie uvedených vzťahov (hlavne znalosť hodnoty y^* a informácie, že táto hodnota predstavuje korektne vytvorený šifrový text). Shoupov protipríklad funguje za predpokladu, že útočník je schopný nejakým spôsobom získať hodnotu s^*.

V prípade RSA-OAEP šifrovacia transformácia (mocnenie mod n) samozrejme nemá konkrétne uvedenú patologickú konštrukciu, pretože napríklad:

- časť odpovedajúca hodnote s^* sa pri šifrovaní nezobrazuje sama na seba;
- nie je známy žiaden efektívny spôsob, akým je možno zo znalosti y^* získať hodnotu s^*.

Vo všeobecnosti však platí, že argumentácia v pôvodnom dôkaze od dvojice Bellare & Rogaway je založená na nedostatočnom predpoklade – jednosmernosti šifrovacej transformácie. Preto môže byť tento dôkaz uplatnený len čiastočne, a to pre demonštráciu bezpečnosti RSA-OAEP voči neadaptívnemu útoku s voľbou

šifrového textu (časť dôkazu odpovedajúca krokom 1-3 a 5 z oddielu 1.2.2).

2.3. Správny dôkaz bezpečnosti RSA-OAEP

V tomto oddiele si predstavíme korektný dôkaz bezpečnosti RSA-OAEP voči adaptívnemu útoku s voľbou šifrového textu, ktorý pochádza od štvorice autorov Fujisaki, Okamoto, Pointcheval a Stern. Vychádzať pritom budeme z publikácie [**16**].

2.3.1. Znalosť otvoreného textu

Definícia (znalosť otvoreného textu). Označme $\mathbb{G}, \mathbb{H}$ zoznamy dvojíc dotaz-odpoveď vznikajúce pri interakcii útočníka $\mathcal{A}$ s náhodnými orákulami G, H. Ďalej označme y^* cieľový šifrový text, ktorý útočník získal od šifrovacieho orákula. Hovoríme, že šifrový systém spĺňuje predpoklad *znalosti otvoreného textu*, ak existuje pravdepodobnostný polynomiálny (vo veľkosti privátneho kľúča) algoritmus nazývaný *extraktor otvoreného textu* $\mathcal{PE}$ s nasledujúcou vlastnosťou – pre každý platný šifrový text $y \neq y^*$ vyprodukovaný útočníkom a jeho odpovedajúci otvorený text x je výraz

$$\mathrm{PA}_{\mathcal{A}} = \Pr[\mathcal{PE}(y, \mathbb{G}, \mathbb{H}, y^*) = x]$$

funkciou, ktorá nie je zanedbateľná.

Poznámka. Predpoklad znalosti otvoreného textu sa v zahraničnej odbornej literatúre najčastejšie označuje pojmom “plaintext awareness”.

Predpoklad znalosti otvoreného textu formálne zachycuje požiadavku, aby útočník nebol schopný vytvoriť platný šifrový text bez znalosti príslušnej správy.

Ak šifrový systém spĺňuje predpoklad znalosti otvoreného textu a zároveň je bezpečný voči útoku s voľbou otvoreného textu, potom je bezpečný aj voči adaptívnemu útoku s voľbou šifrového textu. Formálny dôkaz tohto dôležitého faktu je možné nájsť napríklad v práci [**1**]. Načrtneme si aspoň základnú myšlienku dôkazu. Vezmime do úvahy kryptosystém spĺňujúci predpoklad znalosti otvoreného textu, bezpečný voči útoku s voľbou otvoreného textu a zároveň zraniteľný adaptívnym útokom s voľbou šifrového textu. Uvedená kombinácia vlastností sa však vylučuje. Pri adaptívnom útoku s voľbou šifrového textu môže totiž útočník získať netriviálnu informáciu jedine dotazmi na dešifrovacie orákulum obsahujúcimi platné šifrové texty, ktoré on sám nevytvoril predpísaným spôsobom šifrovania. Také šifrové texty však podľa predpokladu znalosti otvoreného textu útočník nemôže vyprodukovať. Pri strate tejto výhody sa ale útočník fakticky ocitá v scenári útoku s voľbou otvoreného textu. Voči tomuto útoku je pritom uvažovaný kryptosystém bezpečný. Preto musí byť bezpečný aj voči adaptívnemu útoku s voľbou šifrového textu, čím dostávame spor s predpokladom a platnosť tvrdenia zo začiatku odstavca.

Kryptosystém OAEP je podľa svojej definície bezpečný voči útoku s voľbou otvoreného textu (jednosmernou permutáciou sa šifruje správa doplnená o náhodnú hodnotu). Vďaka tomu nám k získaniu bezpečnosti OAEP voči adaptívnemu útoku s voľbou šifrového textu podľa predchádzajúceho odstavca stačí ukázať,

že OAEP splňuje predpoklad znalosti otvoreného textu. Presne takú konštrukciu má správny dôkaz bezpečnosti OAEP (Veta 2.3).

Poznámka. Uvedená definícia predpokladu znalosti otvoreného textu berie do úvahy aj cieľový šifrový text y^*. V prípade, že y^* z tejto definície vynecháme a budeme pracovať s výrazom $\mathrm{WPA}_{\mathcal{A}} = \Pr[\mathcal{PE}(y, \mathbb{G}, \mathbb{H}) = x]$, dostaneme definíciu tzv. slabého predpokladu znalosti otvoreného textu. Splňovanie takého predpokladu spoločne s bezpečnosťou kryptosystému voči útoku s voľbou otvoreného textu implikuje bezpečnosť tohto kryptosystému voči neadaptívnemu útoku s voľbou šifrového textu (myšlienka je analógiou predošlého odstavca). Toto je presne prípad pôvodného dôkazu bezpečnosti RSA-OAEP od dvojice Bellare a Rogaway, kde príslušný extraktor otvoreného textu (simulátor dešifrovacieho orákula) neberie do úvahy možnosť útočníka získať netriviálnu informáciu využitím znalosti cieľového šifrového textu y^*, na základe ktorej môže byť potenciálne schopný vyprodukovať platný šifrový text. Preto tento dôkaz garantuje iba bezpečnosť RSA-OAEP do obdržania cieľového šifrového textu y^* útočníkom, teda v prípade neadaptívneho útoku s voľbou šifrového textu.

2.3.2. Metodika hier

Technika správneho dôkazu je narozdiel od dôkazu dvojice Bellare & Rogaway z minulého oddielu založená na Shoupovej metodike hier, ktorej podrobný popis je možné nájsť v článku [**20**]. Na tomto mieste si v stručnosti vysvetlíme jej podstatu. Cieľom pri dokazovaní bezpečnosti pomocou Shoupovej metodiky hier je vytvorenie postupnosti Hra G_0, Hra G_1, ..., Hra G_n nasledujúcim spôsobom. Hra G_0 je popisom reálneho útoku voči kryptosystému, ktorého bezpečnosť chceme dokázať. Táto bezpečnosť v našom prípade závisí na pravdepodobnosti náhodného javu $b'_{CCA2} = b$ definovaného v oddiele 1.2.2.

Označme tento náhodný jav vyskytujúci sa v Hre G_i ako S_i. Postupnosť Hra G_0, Hra G_0, ..., Hra G_n zostrojíme postupným upravovaním Hry G_0 tak, aby pre všetky $i = 0, \ldots, n-1$ platilo $\Pr[S_i] \approx \Pr[S_{i+1}]$, teda aby rozdiel

$$|\Pr[S_i] - \Pr[S_{i+1}]|$$

bol zanedbateľnou funkciou. Nakoniec je ešte potrebné, aby aj $\Pr[S_n] \approx \frac{1}{2}$. Konštrukciou takej postupnosti dostávame vďaka tomu, že relácia $\approx$ je ekvivalenciou, vzťah $\Pr[S_0] \approx \frac{1}{2}$, čo podľa definície z oddielu 1.2.2 znamená bezpečnosť voči adaptívnemu útoku s voľbou šifrového textu.

Ostáva nám vysvetliť, ako postupným upravovaním Hry G_0 dospejeme až k Hre G_n. Podstata prechodov medzi jednotlivými hrami spočíva v ich vzájomnom odlíšení pomocou nejakej udalosti (chybová udalosť, výmena jednotlivých entít za iné entity s podobnými vlastnosťami, atd.). Hra G_i sa teda od Hry G_{i+1} odlišuje jedine v prípade, že nastane istý definovaný náhodný jav $F_i = F_{i+1} = F$. To znamená, že platí

$$(S_i \ \& \ \neg F) \Leftrightarrow (S_{i+1} \ \& \ \neg F).$$

Uvedenú myšlienku formálne zachycuje nasledujúca Lemma.

Lemma 2.2 (Difference lemma). *Nech E_1, E_2, F_1, F_2 sú náhodné javy, pre ktoré platí:*

$$(\Pr[E_1 \ \& \ \neg F_1] = \Pr[E_2 \ \& \ \neg F_2]) \ \& \ (\Pr[F_1] = \Pr[F_2] = \epsilon).$$

Potom

$$|\Pr[E_1] - \Pr[E_2]| \leq \epsilon.$$

Dôkaz. Výpočtom dostávame:

$$\begin{aligned}|\Pr[E_1] - \Pr[E_2]| &= |\Pr[E_1 \ \& \ \neg F_1] + \Pr[E_1 \ \& \ F_1] - \Pr[E_2 \ \& \ \neg F_2] - \Pr[E_2 \ \& \ F_2]| \\ &= |\Pr[E_1 \ \& \ F_1] - \Pr[E_2 \ \& \ F_2]| = |\Pr[E_1|F_1] \cdot \Pr[F_1] - \Pr[E_2|F_2] \cdot \Pr[F_2]| \\ &= |\Pr[E_1|F_1] - \Pr[E_2|F_2]| \cdot \epsilon \leq \epsilon.\end{aligned}$$

□

2.3.3. Bezpečnosť RSA-OAEP voči CCA2

Vďaka chybe v pôvodnom dôkaze bezpečnosti RSA-OAEP voči adaptívnemu útoku s voľbou šifrového textu, ktorú objavil Shoup, sa zdá byť možnosť tento dôkaz napraviť za udržania pôvodného predpokladu (jednosmernosť šifrovacej permutácie f) nepravdepodobná. Nasledujúci korektný dôkaz preto využíva predpoklad silnejší – čiastočnú jednosmernosť šifrovacej permutácie f.

Definícia (permutácia jednosmerná na častiach definičného oboru). Nech f je jednosmerná šifrovacia permutácia OAEP. Hovoríme, že f je permutácia (τ, ϵ)-*jednosmerná na častiach definičného oboru*, ak pre každého útočníka $\mathcal{A}$, ktorého časová zložitosť je zhora obmedzená výrazom τ, je výraz

$$\text{PD-OW}_f(\mathcal{A}) = \Pr[\mathcal{A}(f(s||t)) = s]$$

pre ľubovoľnú dvojicu (s, t) zhora obmedzený výrazom ϵ.

Definícia (permutácia množinovo jednosmerná na častiach definičného oboru). Nech f je šifrovacia permutácia OAEP, ktorá je jednosmerná na častiach definičného oboru. Hovoríme, že f je permutácia (ℓ, τ, ϵ)-*množinovo jednosmerná na častiach definičného oboru*, ak pre každého útočníka $\mathcal{A}$, ktorého časová zložitosť je zhora obmedzená výrazom τ a ktorého výstupom je množina pozostávajúca z ℓ prvkov, je výraz

$$\text{S-PD-OW}_f(\mathcal{A}) = \Pr[s \in \mathcal{A}(f(s||t))]$$

pre ľubovoľnú dvojicu (s, t) zhora obmedzený výrazom ϵ.

Predchádzajúca definícia formálnym jazykom zahycuje intuitívnu záležitosť – ak je pre ľubovoľného pravdepodobnostného polynomiálneho útočníka úloha odvodiť zo znalosti $y = f(s||t)$ hodnotu s výpočtovo nedosažiteľná, potom je permutácia f jednosmerná na častiach definičného oboru. Množinová varianta je potom len analógiou v prípade, že je výstupom útočníka namiesto jedinej hodnoty množina veľkosti ℓ.
Prejdime k hlavnému tvrdeniu tohto oddielu.

Veta 2.3. *Nech $\mathcal{A}$ je útočník schopný zlomiť kryptosystém OAEP útokom s voľbou šifrového textu s pravdepodobnosťou ϵ, teda nech $\text{Adv}_{\mathcal{A}}^{\text{CCA2}} = \epsilon$. Ďalej nech časová zložitosť $\mathcal{A}$ je zhora obmedzená výrazom τ. Označme počet dotazov $\mathcal{A}$ na dešifrovacie orákulum D a náhodné orákulá G, H po poradí ako q_d, q_g, q_h. Nakoniec označme časovú zložitosť algoritmu počítajúceho funkčnú hodnotu šifrovacej permutácie f ako T_f. Potom platí*

$$\text{S-PD-OW}_f(q_h, \tau') \geq \frac{\epsilon}{2} - \left(\frac{2(q_d + 2)(q_d + 2q_g)}{2^{k_0}} + \frac{3q_d}{2^{k_1}} \right),$$

pričom

$$\tau' \leq \tau + q_g \cdot q_h \cdot (T_f + \mathcal{O}(1)).$$

Ako uvedená veta dokazuje bezpečnosť RSA-OAEP voči adaptívnemu útoku s voľbou šifrového textu? Šifrovacia permutácia f v prípade RSA-OAEP (mocnenie mod n) je (τ, ϵ)-jednosmerná na častiach definičného oboru pre nejaký polynóm τ a nejakú zanedbateľnú funkciu ϵ. Dôkaz tejto skutočnosti možno nájsť napríklad v pojednaní [**9**]. Dôsledkom je podľa definície jednosmernosti na častiach definičného oboru neexistencia pravdepodobnostného polynomiálneho (vo veľkosti privátneho kľúča) útočníka schopného zo znalosti výrazu $y = (s||t)^e \mod n$ odvodiť hodnotu s s pravdepodobnosťou, ktorá nie je zanedbateľná.

Uvažujme teda nejakého útočníka $\mathcal{A}$ schopného atakovať kryptosystém RSA-OAEP s časovou zložitosťou najviac τ' a pravdepodobnosťou úspechu aspoň ϵ, ktorá nie je zanedbateľná. Príslušný výraz $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{CCA2}} = \epsilon$ je preto tiež funkciou, ktorá nie je zanedbateľná. Podľa tvrdenia Vety 2.3 je potom aj výraz $\text{S-PD-OW}_f(q_h, \tau')$ funkciou, ktorá nie je zanedbateľná. Lenže z definície množinovej jednosmernosti na častiach definičného oboru plynie jednoduchou úvahou nerovnosť

$$\frac{\text{S-PD-OW}_f(q_h, \tau')}{q_h} \leq \text{PD-OW}_f(\mathcal{A}),$$

podľa ktorej by aj výraz $\text{PD-OW}_f(\mathcal{A})$ mal byť funkciou, ktorá nie je zanedbateľná. To ale v prípade kryptosystému RSA-OAEP znamená spor s tvrdeniami z predchádzajúceho odstavca. Tým sme za predpokladu platnosti Vety 2.3 vyvrátili existenciu útočníka $\mathcal{A}$ s popísanými vlastnosťami a dostávame tak požadovanú bezpečnosť RSA-OAEP voči adaptívnemu útoku s voľbou šifrového textu.

Pristúpme teda k samotnému dôkazu Vety 2.3.

Dôkaz. V dôkaze sa budeme držať notácie zavedenej v popise scenára adaptívneho útoku s voľbou šifrového textu z oddielu 1.2.2. Všetky hodnoty vzťahujúce sa k cieľovému šifrovému textu y^* budeme rovnako označovať hviezdičkou, máme teda $r^* = H(s^*) \oplus t^*$, $G(r^*) = s^* \oplus (m_b||0^{k_1})$.

Extraktor otvoreného textu. Aby sme dokázali bezpečnosť voči adaptívnemu útoku s voľbou šifrového textu, musíme byť schopní simulovať odpovede na dotazy na dešifrovacie orákulum. K tomu účelu nám poslúži extraktor otvoreného textu $\mathcal{PE}$. Jeho program definujeme rovnako, ako program simulátora z pôvodného dôkazu bezpečnosti RSA-OAEP. Extraktor otvoreného textu $\mathcal{PE}$ teda pri obdržaní dotazu obsahujúceho šifrový text $y = f(s||t)$ postupuje nasledovne:

- Buď boli príslušné hodnoty s, r v priebehu výpočtu dotázané orákul H, G. Označme zoznam všetkých dvojíc dotázaných hodnôt a príslušných odpovedí orákul H, G po poradí ako $\mathbb{H}$, $\mathbb{G}$. V tom prípade prejde $\mathcal{PE}$ všetky uložené dvojice $(\gamma, G_\gamma) \in \mathbb{G}$ a $(\delta, H_\delta) \in \mathbb{H}$. Pre každú dvojicu spočíta výrazy $\sigma = \delta$, $\theta = \gamma \oplus H_\delta$, $\mu = G_\gamma \oplus \delta$. Ďalej overí, či platí rovnosť $y = f(\sigma, \theta)$. Ak áno, skontroluje, či posledných k_1 bitov μ tvoria samé nuly. V prípade, že boli nájdené hodnoty spĺňajúce všetky uvedené vzťahy, $\mathcal{PE}$ vráti ako svoju odpoveď prvých n bitov μ. Jednoznačnosť nájdenej odpovede plynie z predpokladu, že f je permutácia.

- Alebo vráti $\mathcal{PE}$ ako svoju odpoveď hlášku `zlyhanie`. Ukážeme si, že v tomto prípade existuje len zanedbateľná pravdepodobnosť (rovná $1/2^{k_0}$), že dotázaný šifrový text y bol vytvorený korektne a vyhovuje predpísanej štruktúre otvoreného textu pre OAEP (teda že posledných k_0 bitov tvoria samé nuly).

Hra G_0. Scenár hry G_0 je zhodný s reálnym scenárom adaptívneho útoku s voľbou šifrového textu popísaným v oddiele 1.2.2. Udalosť $b'_{CCA2} = b$ označme ako S_0. Podľa definície výhody útočníka nastáva táto udalosť s pravdepodobnosťou

$$\Pr[S_0] = \frac{1}{2} + \frac{\epsilon}{2}.$$

Rovnaké udalosti budeme v nasledujúcich hrách $\mathrm{G_i}$ označovať ako S_i.

Hra G_1. V tejto hre pristúpime k prvej simulácii reálneho útoku. Tú získame nahradením všetkých odpovedí náhodných orákul G, H odpoveďami volenými náhodne a uniformne. Podrobný program simulácie zachycuje Tabuľka 2.1. Keďže podľa definície náhodného orákula sa distribúcia nových odpovedí nezmení, platí rovnosť

$$\Pr[S_0] = \Pr[S_1].$$

Hra G_2. V tejto hre zmeníme oproti schéme z Tabuľky 2.1 pravidlo **Chall-Hash** nasledovne:

| • **Pravidlo Chall-Hash$^{(2)}$:**
| Zvoľ náhodne a uniformne r^+, g^+, polož $r^* = r^+$, $g^* = g^+$
| a spočítaj $s^* = M^* \oplus g^+$, $h^* = H(s^*)$, $t^* = r^+ \oplus h^*$.

Hry G_1 a G_2 sú nerozlíšiteľné až na udalosť, kedy sa buď útočník alebo dešifrovacie orákulum dotáže orákula G hodnotou r^*. Označme tento jav ako $\mathrm{Ask}_{\mathrm{G}_2}$ a rovnaký jav vyskytujúci sa v i-tej hre ako $\mathrm{Ask}_{\mathrm{G_i}}$. Platí teda:

$$\Pr[\mathrm{Ask}_{\mathrm{G}_1}] = \Pr[\mathrm{Ask}_{\mathrm{G}_2}]$$

$$\Pr[S_1 \ \& \ \neg\mathrm{Ask}_{\mathrm{G}_1}] = \Pr[S_2 \ \& \ \neg\mathrm{Ask}_{\mathrm{G}_2}]$$

Dôsledkom uvedených vzťahov je podľa Lemmy 2.2 nerovnosť:

$$|\Pr[S_2] - \Pr[S_1]| \leq \Pr[\mathrm{Ask}_{\mathrm{G}_2}]$$

V tejto hre sme určili hodnoty r^+, g^+ nezávisle na sebe, teda hodnota g^+ nie je definovaná ako $G(r^+)$. Napriek tomu sú obe hodnoty použité pri výpočte cieľového šifrového textu y^*. Keďže y^* je po uvedenej zmene programu vytvorený náhodne, nepriateľ stráca svoju výhodu (tzn. $\epsilon = 0$) a preto je:

$$\Pr[S_2] = \frac{1}{2}.$$

Hra G_3. V tejto hre sa začneme zaoberať simuláciou dešifrovacieho orákula, z ktorého chceme postupne dostať extraktor otvoreného textu. Program D upravíme tak, že bude odmietať všetky šifrové texty y, pre ktoré príslušná hodnota r nebola v priebehu útoku dotázaná náhodného orákula G. To znamená zmenu pravidla **Decrypt-SnoR** nasledovne:

| • **Pravidlo Decrypt-SnoR$^{(3)}$:**
| $g = G(r)$, $M = 1^k$.

Uplatnenie daného pravidla vedie k odmietnutiu šifrového textu kvôli jeho nezhode so štruktúrou definovanou OAEP (zakončenie šifrového textu k_1 nulovými bitmi). Hra G_3 sa oproti tej predchádzajúcej líši v prípade, kedy nastane nasledujúca udalosť – dotázaný šifrový text y je platný a zároveň príslušná hodnota r nebola dotázaná náhodného orákula G – označme túto udalosť ako E_3. V tomto

Program náhodného orákula G:
Dotaz $G(r)$:
Ak v $\mathbb{G}$ existuje záznam (r, g), vráť g.
Inak zvoľ náhodne a uniformne $g \in \{0,1\}^{k-k_0}$, ulož záznam (r, g) do $\mathbb{G}$ a vráť g.

Program náhodného orákula H:
Dotaz $H(s)$:
Ak v $\mathbb{H}$ existuje záznam (s, h), vráť h.
Inak zvoľ náhodne a uniformne $h \in \{0,1\}^{k_0}$, ulož záznam (s, h) do $\mathbb{H}$ a vráť h.

Program dešifrovacieho orákula D:
Dotaz $D(y)$:
- **Pravidlo Decrypt-Init$^{(1)}$:**
  Spočítaj $(s||t) = f^{-1}(y)$.

Vyhľadaj dvojicu (s, h) v $\mathbb{H}$:
Ak existuje, spočítaj $r = t \oplus h$
Vyhľadaj dvojicu (r, g) v $\mathbb{G}$:
Ak existuje, vykonaj
- **Pravidlo Decrypt-SR$^{(1)}$:**
  $h = H(s)$, $r = t \oplus h$
  $g = G(r)$, $M = s \oplus g$

Inak vykonaj
- **Pravidlo Decrypt-SnoR$^{(1)}$:**
  $h = H(s)$, $r = t \oplus h$
  $g = G(r)$, $M = s \oplus g$

Inak vykonaj
- **Pravidlo Decrypt-noS$^{(1)}$:**
  $h = H(s)$, $r = t \oplus h$
  $g = G(r)$, $M = s \oplus g$

Ak je posledných k_1 bitov M nulových, vráť m pozostávajúce z prvých n bitov M.
Inak vráť hlášku `zlyhanie`.

Program výroby cieľového šifrového textu:
Na základe obdržania dvojice správ (m_0, m_1) zvoľ náhodne a uniformne $b \in \{0,1\}$ a spočítaj $m^* = m_b$, $M^* = m^*||0^{k_1}$.
- **Pravidlo Chall-Hash$^{(1)}$:**
  Zvoľ náhodne a uniformne r^* a spočítaj
  $g^* = G(r^*)$, $s^* = M^* \oplus g^*$
  $h^* = H(s^*)$, $t^* = r^* \oplus h^*$
- **Pravidlo Chall-Output$^{(1)}$:**
  Spočítaj $y^* = f(s^*||t^*)$ a vráť y^*.

TABUĽKA 2.1. Simulácia scenára útoku CCA2 voči OAEP v hre G_1

prípade je teda podľa upraveného predpisu najskôr uskutočnený dotaz na hodnotu $G(r)$. Ale keďže táto hodnota má podľa definície náhodného orákula uniformnú distribúciu na $\{0,1\}^{k-k_0}$, je posledných k_1 bitov hodnoty $s \oplus G(r)$ nulových (ako vyžaduje predpísaná štruktúra otvoreného textu OAEP) len s pravdepodobnosťou

$1/2^{k_1}$. Ak pravdepodobnosti sčítame cez všetky dotazy na dešifrovacie orákulum, ktorých je q_d, dostávame:

$$\Pr[E_3] \leq \frac{q_d}{2^{k_1}}.$$

Ďalej platí:

$$\Pr[\mathrm{AskG}_2 \ \& \ \neg E_3] = \Pr[\mathrm{AskG}_3 \ \& \ \neg E_3].$$

A preto podľa Lemmy 2.2 dostávame:

$$|\Pr[\mathrm{AskG}_3] - \Pr[\mathrm{AskG}_2]| \leq \frac{q_d}{2^{k_1}}.$$

Dôvod, prečo v novom pravidle **Decrypt-SnoR**$^{(3)}$ ostáva volanie náhodného orákula $g = G(r)$ aj napriek tomu, že výsledná hodnota $M = 1^k$ je pevne definovaná, spočíva v uložení dvojice (r, g) do $\mathbb{G}$. Dotazovaná hodnota r by totiž mohla byť rovná hodnote r^*, čo by ovplyvnilo pravdepodobnosť výskytu javu AskG_3.

Hra G_4. V tejto hre ďalej upravíme program dešifrovacieho orákula tak, že D bude odmietať všetky šifrové texty y, pre ktoré príslušná hodnota s nebola v priebehu útoku dotázaná náhodného orákula H. To sa prejaví na pravidle **Decrypt-noS** nasledovne:

- **Pravidlo Decrypt-noS**$^{(4)}$**:**
  $h = H(s)$, $r = t \oplus h$
  $g = G(r)$, $M = 1^k$.

Hra G_4 sa oproti tej predchádzajúcej líši v prípade, kedy je dotázaný šifrový text y platný a zároveň príslušná hodnota s nebola dotázaná náhodného orákula H. Keď nastane volanie pravidla **Decrypt-noS**$^{(4)}$, nezistíme už, či bola príslušná hodnota $r = H(s) \oplus t$, ktorá má uniformnú distribúciu na $\{0,1\}^{k_0}$, dotázaná. Preto musíme rozlíšiť dva prípady:

- pravdepodobnosť, že v priebehu útoku hodnota r dotázaná bola, je najviac $(q_g+q_d)/2^{k_0}$ (na príslušné r sa útočník mohol dotázať buď priamo náhodného orákula G alebo nepriamo prostredníctvom dešifrovacieho orákula D).
- ak hodnota r dotázaná ešte nebola, tak podľa úvahy z predchádzajúcej hry je pravdepodobnosť, že y je korektný šifrový text, rovná $1/2^{k_1}$. Opäť sčítaním cez všetky volania dešifrovacieho orákula a použitím Lemmy 2.2 dostávame:

$$|\Pr[\mathrm{AskG}_4] - \Pr[\mathrm{AskG}_3]| \leq \frac{q_d(q_g + q_d)}{2^{k_0}} + \frac{q_d}{2^{k_1}}.$$

Hra G_5. V tejto hre upravíme pravidlo **Decrypt-noS** nasledovne:

- **Pravidlo Decrypt-noS**$^{(5)}$**:**
  $h = H(s)$, $M = 1^k$.

Táto úprava sa zakladá na úvahe, že nech je v predchádzajúcej hre pri aplikácii pravidla **Decrypt-noS** hodnota h akákoľvek, hodnota $M = 1^k$ je pevne daná, a tak ostávajú hodnoty g, h útočníkovi skryté (nemôže ich odvodiť na základe vráteného výstupu, ktorým je len hláška `zlyhanie`). Modifikované pravidlo **Decrypt-noS** teda vynechá dotaz na hodnotu r, čo znamená vypadnutie páru (r, g) zo zoznamu $\mathbb{G}$. Tento krok môže mať nasledujúce dôsledky:

- ak dešifrovacie orákulum neskôr obdrží ako vstup šifrový text y', pre ktorý je príslušná hodnota r' zhodná s hodnotou r (toto r sa vďaka zmene pravidla **Decrypt-noS** v tejto hre nedefinovalo), je v aktuálnej hre G_5 volanie pravidla **Decrypt-SR** nahradené volaním pravidla **Decrypt-SnoR**. Preto je príslušné $g' = g$ vďaka aplikácii pravidla **Decrypt-SnoR** práve definované

(v predchádzajúcej hre by definované už bolo), čím sa dostávame do situácie z hry G_3. Preto pravdepodobnosť, že M' vyhovuje predpísanej štruktúre OAEP, je rovná $1/2^{k_1}$.

- hodnota $r = H(s) \oplus t$, ktorá sa vďaka zmene pravidla **Decrypt-noS** v tejto hre nedefinovala, mohla byť zhodná s hodnotou r^*. A keďže $H(s)$ má uniformnú distribúciu na $\{0,1\}^{k_0}$, je pravdepodobnosť tohto javu $1/2^{k_0}$.

Opäť sčítaním cez všetky volania dešifrovacieho orákula a použitím Lemmy 2.2 dostávame:

$$|\Pr[\mathrm{AskG}_5] - \Pr[\mathrm{AskG}_4]| \leq \frac{q_d}{2^{k_0}} + \frac{q_d}{2^{k_1}}.$$

Hra G_6. V tejto hre upravíme pravidlo **Decrypt-noS** nasledovne:

- **Pravidlo Decrypt-noS$^{(6)}$:**
  $M = 1^k$.

Modifikované pravidlo **Decrypt-noS** teda oproti minulej hre vynechá aj dotaz na hodnotu s, čo znamená vypadnutie páru (s,h) zo zoznamu $\mathbb{H}$. Dôsledky tohto kroku sa môžu prejaviť v prípade, ak dešifrovacie orákulum neskôr obdrží ako vstup šifrový text y', pre ktorý je príslušná hodnota s' zhodná s hodnotou s (toto s vďaka zmene pravidla **Decrypt-noS** v hre G_6 ešte definované nebolo – narozdiel od hry G_5). Konkrétne je v tomto prípade:

- ak by hodnota r' nebola dotázaná, je volanie pravidla **Decrypt-SnoR** nahradené volaním pravidla **Decrypt-noS**. To znamená vynechanie dotazu na hodnotu r', ktorá potenciálne môže byť zhodná s hodnotou r^*. Pravdepodobnosť tohto javu je $1/2^{k_0}$.
- ak by hodnota r' dotázaná bola, je volanie pravidla **Decrypt-SR** nahradené volaním pravidla **Decrypt-noS**. Pravdepodobnosť, že platí rovnosť $r' = t' \oplus h'$, pričom hodnota h' má uniformnú distribúciu na $\{0,1\}^{k_0}$ a vďaka realizovanej zmene ešte nebola definovaná (a kvôli zmenám v tejto hre už simulátorom dešifrovacieho orákula ani definovaná byť nemôže), je teda najviac $q_g/2^{k_0}$.

Opäť sčítaním cez všetky volania dešifrovacieho orákula a použitím Lemmy 2.2 dostávame:

$$|\Pr[\mathrm{AskG}_6] - \Pr[\mathrm{AskG}_5]| \leq \frac{q_d \cdot q_g}{2^{k_0}} + \frac{q_d}{2^{k_0}}.$$

Navyše v tejto hre zmeníme aj pravidlo **Decrypt-SR**, ktoré sa volá v prípade, že obe hodnoty r, s už boli dotázané, nasledovne:

- **Pravidlo Decrypt-SR$^{(6)}$:**
  $M = s \oplus g$.

Uvedená zmena nemá žiaden vplyv na pravdepodobnosti javov, ktorými sa v dôkaze zaoberáme.

Hra G_7. Táto hra je kľúčovou v celom dôkaze. Pravidlo **Chall-Hash** upravíme nasledovne:

- **Pravidlo Chall-Hash$^{(7)}$:**
  Zvoľ náhodne a uniformne r^+, s^+, h^+ a spočítaj
  $s^* = s^+$, $t^* = r^+ \oplus h^+$.

Dôsledkom uvedenej úpravy je vzájomná nezávislosť hodnôt s^+ a h^+. Teda naďalej už neplatí rovnosť medzi výrazmi h^+ a $H(s^+)$. Nezávislým definovaním s^+ dostávame tiež obmedzujúcu požiadavku na hodnotu g^+, pretože $g^+ = M^* \oplus s^+$. Ale to nevadí, lebo hodnota g^+ by sa ajtak využívala iba pre vytvorenie hodnoty s^+ podľa vzťahu $s^+ = M^* \oplus g^+$.

Označme jav, kedy sa útočník dotáže na hodnotu s^* náhodného orákula H, ako AskH_7. Na tomto mieste je dôležité uvedomiť si rozdiel medzi javmi AskG a AskH. Prvý z nich nastáva, keď je hodnota r^* dotázaná útočníkom alebo simulátorom dešifrovacieho orákula, pričom jav AskH môže nastať jedine v prípade, kedy bola hodnota s^* dotázaná útočníkom. Dôvodom je, že postupne upravovaný simulátor dešifrovacieho orákula už v hre G_7 vo svojom programe neobsahuje žiadne volanie náhodného orákula H.

Práve uvedená odlišnosť vo vnímaní udalostí AskG a AskH, ktorá je dôsledkom špecifického prístupu a myšlienkovej konštrukcie dôkazu, zohráva v celom dôkaze zásadnú úlohu. Táto odlišnosť taktiež predstavuje hlavný rozdiel oproti nesprávnemu dôkazu z minulého oddielu, v ktorom sa medzi volaniami jendotlivých náhodných orákul G a H pri simulácii útoku nijak nerozlišovalo – a ako sa v konečnom dôsledku ukázalo, tento prístup je vo všeobecnosti možné považovať za chybný.

Pokračujme teda ďalej v správnom dôkaze. Hra G_7 sa vďaka zrušeniu vzájomnej väzby hodnôt s^+ a h^+ oproti predchádzajúcej hre líši v prípade, ak buď nastane jav AskH_7 alebo ak je hodnota s^* použitá dešifrovacím orákulom. Rozoberme si samostatne dôsledky prípadu, kedy je hodnota s^* použitá simulátorom dešifrovacieho orákula. Nech teda $y = f(s^*||t)$ je platný šifrový text, potom inkonzistencia nastáva, ak:

- príslušná hodnota r bola dotázaná a nastalo volanie pravidla **Decrypt-SR**. Platí $r = t \oplus H(s^*) = t \oplus t^* \oplus r^+$, kde hodnota r^+ má podľa upraveného pravidla **Chall-Hash** uniformnú distribúciu na $\{0,1\}^{k_0}$ a teda neplatí rovnosť $H(s^*) = t^* \oplus r^+$. Preto pravdepodobnosť, že hodnota r splňujúca uvedený vzťah bola dotázaná náhodného orákula G buď priamo útočníkom alebo prostredníctvom dešifrovacieho orákula, je najviac $(q_g + q_d)/2^{k_0}$.
- príslušná hodnota r dotázaná nebola a nastalo volanie pravidla **Decrypt-SnoR**, pričom zároveň nastala rovnosť $r = r^+$ a teda nasleduje volanie $G(r) = G(r^+)$ a uloženie dvojice $(r^+, G(r^+))$ do zoznamu $\mathbb{G}$. Keďže hodnota r^+ má uniformnú distribúciu na $\{0,1\}^{k_0}$, je pravdepodobnosť tohto javu rovná $1/2^{k_0}$.

Opäť sčítaním cez všetky volania dešifrovacieho orákula a použitím Lemmy 2.2 dostávame:

$$|\Pr[\text{AskG}_7] - \Pr[\text{AskG}_6]| \leq \Pr[\text{AskH}_7] + \frac{q_d(q_g + q_d)}{2^{k_0}} + \frac{q_d}{2^{k_0}}.$$

Ďalej platí, že hodnota $r^* = t^* \oplus h^+$ je z pohľadu útočníka zviazaná s nezávisle a náhodne volenou hodnotou h^+, ktorá nie je rovná $H(s^+)$. Hodnotu r^* preto útočník nemôže získať inak, než priamym dotazom na náhodné orákulum G alebo sprostredkovane cez simulátor dešifrovacieho orákula:

$$\Pr[\text{AskG}_7] \leq \frac{(q_g + q_d)}{2^{k_0}}.$$

Vďaka uvedenému hornému odhadu a nerovnostiam, ktoré sme odvodili v predchádzajúcich hrách, je pomocou aritmetických úprav možné obdržať:

$$\Pr[\text{AskG}_2] \leq \frac{3q_d}{2^{k_1}} + \frac{(2q_d + 1)(q_g + q_d)}{2^{k_0}} + \frac{q_d(q_g + 3)}{2^{k_0}} + \Pr[\text{AskH}_7].$$

Hra G_8. V tejto hre upravíme pravidlo **Chall-Hash** nasledovne:

- **Pravidlo Chall-Hash$^{(8)}$:**

| Zvoľ náhodne a uniformne s^+, t^+ a polož
| $s^* = s^+$, $t^* = t^+$.

Vďaka uniformným distribúciám všetkých prvkov, s ktorými sa narába pri volaní pravidla **Chall-Hash**, sa v hre G_8 oproti predchádzajúcej hre žiadna zo študovaných pravdepodobností nezmení.

Hra G_9. V tejto hre upravíme pravidlo **Chall-Hash** nasledovne:

| • **Pravidlo Chall-Hash$^{(9)}$:**
| <Prázdny program>

Ďalej upravíme pravidlo **Chall-Output** nasledovne:

| • **Pravidlo Chall-Output$^{(9)}$:**
| Zvoľ náhodne a uniformne y^+ a polož
| $y^* = y^+$.

Cieľový šifrový text je teda v tejto hre definovaný nezávisle na akomkoľvek výpočte alebo inej hodnote. Jeho distribúcia ale ostáva rovnaká (vďaka tomu, že permutácia f je náhodná) a preto sa v hre G_9 oproti predchádzajúcej hre žiadna zo študovaných pravdepodobností nezmení. Na druhej strane, definovanie hodnoty y^+ implicitne určuje príslušné hodnoty s^+, t^+ (vďaka tomu, že zobrazenie f je permutácia). Preto sa na hodnotu $y^+ = f(s^+||t^+)$ môžeme pozerať ako na náhodne určenú výzvu, ktorú chce útočník čiastočne invertovať, teda nájsť k nej odpovedajúcu hodnotu s^+.

Hra G_{10}. V poslednej hre upravíme pravidlo **Decrypt-SnoR** nasledovne:

| • **Pravidlo Decrypt-SnoR$^{(10)}$:**
| $M = 1^k$.

Táto zmena síce znamená vynechanie dotazu r na náhodné orákulum G, nás však v hre G_{10} už pravdepodobnosť javu $\mathrm{Ask}G_{10}$ nezaujíma.

Je vidieť, že sme dospeli k požadovanej konštrukcii extraktora otvoreného textu, ktorým je zostrojený simulátor dešifrovacieho orákula. Ostáva nám ukázať platnosť tvrdenia dokazovanej Vety. To však plynie z jednoduchého pozorovania:

$$\Pr[\mathrm{Ask}H_{10}] \leq \text{S-PD-OW}_f(q_h, \tau').$$

Jav $\mathrm{Ask}H_{10}$ totiž nastane, ak aspoň jeden z celkového počtu q_h útočníkových dotazov na náhodné orákulum H bude obsahovať hodnotu s^+, ktorá je zároveň čiastočným inverzom k cieľovému šifrovému textu $y^+ = f(s^+||t^+)$. Summa summarum je teda

$$\Pr[\mathrm{Ask}G_2] \leq \frac{3q_d}{2^{k_1}} + \frac{(2q_d+1)(q_g+q_d)}{2^{k_0}} + \frac{q_d(q_g+3)}{2^{k_0}} + \Pr[\mathrm{Ask}H_{10}]$$

$$\leq \frac{3q_d}{2^{k_1}} + \frac{(2q_d+1)(q_g+q_d)}{2^{k_0}} + \frac{q_d(q_g+3)}{2^{k_0}} + \text{S-PD-OW}_f(q_h, \tau')$$

a podľa úvah z hry G_2 platí tiež

$$\frac{\epsilon}{2} = |\Pr[S_2] - \Pr[S_1]| \leq \Pr[\mathrm{Ask}G_2],$$

odkiaľ použitím jednoduchých aritmetických úprav dostávame priamo tvrdenie dokazovanej Vety:

$$\text{S-PD-OW}_f(q_h, \tau') \geq \frac{\epsilon}{2} - \left(\frac{2(q_d+2)(q_d+2q_g)}{2^{k_0}} + \frac{3q_d}{2^{k_1}}\right).$$

Časová zložitosť. Aby sme ukázali odhad pre časovú zložitosť útoku, upravíme dátovú štruktúru v skonštruovanom extraktore otvoreného textu. Namiesto ukladania celých zoznamov $\mathbb{G}$, $\mathbb{H}$ bude ukladaná iba dátová štruktúra pozostávajúca z usporiadaných pätíc tvaru

$$(\gamma, G_\gamma, \delta, H_\delta, y),$$

kde $(\gamma, G_\gamma) \in \mathbb{G}$ a $(\delta, H_\delta) \in \mathbb{H}$. Pre každú takú dvojicu (δ, γ) sa potom spočíta $\sigma = \delta$, $\theta = \gamma \oplus H_\delta$, $\mu = G_\gamma \oplus \delta$ a $y = f(\sigma, \theta)$. Ak posledných k_1 bitov μ tvoria samé nuly, je pätica $(\gamma, G_\gamma, \delta, H_\delta, y)$ uložená. Časová zložitosť, ktorú si udržiavanie takejto dátovej štruktúry vyžaduje, môže byť zhora odhadnutá výrazom

$$q_g \cdot q_h \cdot (T_f + \mathcal{O}(1)),$$

kde T_f označuje časovú zložitosť algoritmu počítajúceho funkčnú hodnotu f. Výhodou definovanej dátovej štruktúry je schopnosť extraktora otvoreného textu vďaka tejto štruktúre k obdržanému y vrátiť príslušný otvorený text v konštantnom čase (všetko je predpočítané dopredu a dešifrovanie y je záležitosťou vyhľadavánia v uloženej dátovej štruktúre podľa hodnoty y). Podarilo sa nám teda zhora odhadnúť časovú zložitosť simulovaného útoku s využitím extraktora otvoreného textu. Tento horný odhad je polynomiálnou funkciou pôvodnej časovej zložitosti a tak konečne dostávame požadované – bezpečnosť OAEP voči adaptívnemu útoku s voľbou šifrového textu za predpokladu polynomiálne obmedzenej výpočtovej sily útočníka a množinovej jednosmernosti na častiach definičného oboru v prípade použitej permutácie f.

□

3
Praktické aspekty bezpečnosti RSA-OAEP

3.1. Problém nedostatočne priliehavej redukcie

Najprv si rozoberme konkrétne dôsledky redukcie v dôkaze z predchádzajúceho oddielu z hľadiska časovej zložitosti. Vychádzať pri tom budeme z publikácie [**16**].

Dôkaz uvedený v predchádzajúcom oddiele je platný pre OAEP všeobecne, tj. s použitím ľubovoľnej šifrovacej transformácie, ktorá je jednosmerná na častiach definičného oboru. Takou transformáciou je i funkcia RSA (umocňovanie na e modulo n). Aplikácia dôkazu bezpečnosti voči adaptívnemu útoku s voľbou šifrového textu na prípad kryptosystému RSA-OAEP využíva techniku redukcie mreží a je možné ju nájsť v článku [**9**]. Podstatným pre túto prácu je fakt, že pre RSA-OAEP je podľa uvedeného článku v prípade zraniteľnosti adaptívnym útokom s voľbou šifrového textu pravdepodobnosť vyriešenia problému RSA (tj. nájdenia e-tej odmocniny $\mod n$) zdola obmedzená takto

$$\epsilon' \geq \epsilon^2 - 2\epsilon \cdot (\frac{2q_d q_g + q_d + q_g}{2^{k_0}} + \frac{2q_d}{2^{k_1}} + \frac{32}{2^{k-2k_0}}),$$

pričom príslušná časová zložitosť je zhora obmedzená nasledovne

$$t' \leq 2t + q_h(q_h + 2q_g) \times \mathcal{O}(k^3),$$

kde k označuje bitovú dĺžku RSA modulu. Uvedené odhady časovej zložitosti t' a pravdepodobnosti ϵ' sa vzťahujú na simulátor, ktorý k dosiahnutiu cieľa využije útočníka ako svoj podprogram. Výrazy t a ϵ náležia útočníkovi atakujúcemu kryptosystém RSA-OAEP v zmysle dôkazu z predchádzajúceho oddielu.

Aby sme si ukázali, aké praktické dôsledky pre bezpečnosť RSA-OAEP má dôkaz z predchádzajúceho oddielu, musíme si stanoviť nejaké referenčné merítko. V súčasnosti posledným známym RSA modulom, ktorý bol faktorizovaný, je RSA-200. Faktorizácia bola realizovaná metódou všeobecného číselného sita, ktorá je pre podobné prípady metódou momentálne najrýchlejšou známou. RSA-200 má dĺžku 200 číslic v dekadickom zápise, čo predstavuje dĺžku 663 bitov. Je žiadúce, aby bol RSA modul jednak čo najkratší, aby operácie v príslušnom okruhu $\mathbb{Z}_n$ trvali čo najkratšie a zároveň čo najdlhší, aby bola jeho faktorizácia čo najťažšia. Uvedené podmienky je v praxi potrebné vyvážiť. V súčasnosti sa preto najčastejšie používajú RSA moduly o bitovej dĺžke 1024 bitov. To je stále ďaleko za hranicou 663 bitov a tým aj reálneho rizika faktorizácie, a preto sa používanie

RSA modulov dĺžky aspoň 1024 bitov považuje v praxi za bezpečné. Poznamenajme, že momentálne nie je známa metóda, ktorá by vo všeobecnom prípade viedla k invertovaniu RSA funkcie (a tým k rekonštrukcii správy zo šifrového textu RSA-OAEP) efektívnejšie, než faktorizácia modulu.

Chceme odvodiť približnú veľkosť RSA modulu tak, aby sme našli hranicu, od ktorej začína mať dôkaz správnosti praktickú výpovednú hodnotu. Ako referenčné merítko si podľa predchádzajúceho odstavca stanovíme časovú náročnosť faktorizácie metódou všeobecného číselného sita. Tá je daná výrazom

$$\mathcal{O}\left(e^{1{,}9(\ln n)^{1/3}(\ln\ln n)^{2/3}}\right),$$

kde n označuje RSA modul. Konkrétne platí:

- pre RSA modul dĺžky 1024 bitov (tj. pre $n \doteq 2^{10}$) je časová náročnosť jeho faktorizácie rádovo 2^{80} krokov;
- pre RSA modul dĺžky 2048 bitov (tj. pre $n \doteq 2^{11}$) je časová náročnosť jeho faktorizácie rádovo 2^{111} krokov;
- pre RSA modul dĺžky 4096 bitov (tj. pre $n \doteq 2^{12}$) je časová náročnosť jeho faktorizácie rádovo 2^{149} krokov.

Aby sme dostali hľadanú hranicu veľkosti pre RSA modul, potrebujeme získané hodnoty porovnať s hodnotami plynúcimi zo vzťahov pre časovú zložitosť t' a pravdepodobnosť úspechu ϵ' simulátora snažiaceho sa invertovať RSA funkciu využitím útočníka atakujúceho kryptosystém RSA-OAEP. Pretože bezpečnosť kryptosystému RSA-OAEP chceme analyzovať čo najviac prakticky, položíme $\epsilon = 1$, tj. v tomto prípade uvažovaný útočník atakujúci RSA-OAEP adaptívnym útokom s voľbou šifrového textu uspeje s istotou. Po dosadení hodnoty $\epsilon = 1$ do odhadu pre ϵ' zistíme, že hodnota odhadu bude veľmi blízka 1 (dôvodom je, že člen odhadu obsahujúci ϵ v prvej mocnine je v praxi zanedbateľný vďaka vysokým hodnotám k_0 a k_1). Takto dostávame úspešnosť simulátora pri invertovaní RSA funkcie na úrovni istoty, čiže podobnú ako v prípade faktorizácie RSA modulu metódou všeobecného číselného sita. Tým pádom sa môžeme presunúť k porovnaniu časovej náročnosti. Pripomeňme si odhad z úvodu oddielu:

$$t' \leq 2t + q_h(q_h + 2q_g) \times \mathcal{O}(k^3).$$

Aby bol tento odhad prakticky použiteľný, musíme si stanoviť adekvátny odhad výpočtových možností útočníka, s ktorými sa v ňom pracuje. Pointcheval vo svojej publikácii uvádza odhady:

- maximálny celkový čas útoku je $t \leq 2^{75}$ krokov;
- maximálny počet dotazov na jedno orákulum je $q_x \leq 2^{55}$, kde $x \in \{g, h\}$.

Dosadením týchto čísel dostávame nasledujúce údaje:

- pre RSA modul dĺžky 1024 bitov (tj. pre $k = 2^{10}$) je časová náročnosť invertovania RSA funkcie simulátorom rádovo $t' \leq 2^{143}$ krokov;
- pre RSA modul dĺžky 2048 bitov (tj. pre $k = 2^{11}$) je časová náročnosť invertovania RSA funkcie simulátorom rádovo $t' \leq 2^{146}$ krokov;
- pre RSA modul dĺžky 4096 bitov (tj. pre $k = 2^{12}$) je časová náročnosť invertovania RSA funkcie simulátorom rádovo $t' \leq 2^{149}$ krokov.

Porovnaním získaných hodnôt s odpovedajúcimi hodnotami pre faktorizáciu RSA modulu metódou všeobecného číselného sita vidíme, že odhad pre časovú náročnosť simulátora invertujúceho RSA funkciu využitím útočníka úspešne atakujúceho RSA-OAEP je prakticky použiteľný až pre RSA moduly dĺžky aspoň 4096 bitov. Pre RSA moduly kratšej dĺžky totiž nemá v podstate žiadnu výpovednú hodnotu, čo si ukážeme na nasledujúcom príklade. Vezmime do úvahy RSA modul dĺžky 1024 bitov. Získané odhady hovoria, že ak existuje útočník úspešne atakujúci kryptosystém RSA-OAEP, potom simulátor využívajúci útočníka úspešne atakujúceho RSA-OAEP ako svoj podprogram dokáže invertovať RSA funkciu v čase nanajvýš 2^{143}. Zároveň je však známy spôsob, ktorý vedie k invertovaniu RSA funkcie efektívnejšie – faktorizácia RSA modulu metódou všeobecného číselného sita v čase rádovo 2^{80} krokov. Preto je odvodené tvrdenie ako argument demonštrujúci bezpečnosť RSA-OAEP voči adaptívnemu útoku s voľbou šifrového textu v tomto prípade nepoužiteľné. Situácia sa mení až pri veľkosti RSA modulu aspoň 4096 bitov. Od tejto hranice je totiž invertovanie RSA funkcie simulátorom efektívnejšie než faktorizácia RSA modulu číselným sitom, ktoré je momentálne najefektívnejšou známou metódou pre invertovanie RSA funkcie.

Ukázali sme si teda, že dôkaz bezpečnosti OAEP z minulého oddielu a jeho aplikácia na kryptosystém RSA-OAEP síce má konkrétny reálny dopad, avšak vzhľadom k súčasne používaným dĺžkám RSA modulov (1024 bitov) sa skôr jedná o dopad naplno použiteľný až v budúcnosti, keď sa štandardom stanú RSA moduly dĺžky aspoň 4096 bitov. Tento dôkaz je ale v každom prípade významný už samotnou svojou existenciou, ktorá výrazne podporuje spoľahlivosť konštrukcie OAEP.

3.2. Problém nekvalitného generátora náhodných čísel

Bezpečnosť RSA-OAEP závisí pri praktickom nasadení na množstve detailov. Jedným z nich, ako ukázal Brown vo svojej práci [**6**], je aj dostatočne kvalitný generátor náhodných čísel. Pri kódovaní správy m podľa schémy OAEP totiž do výpočtu vstupuje náhodná hodnota r. Ak je r generované nedostatočne náhodne a ak navyše šifrovacia transformácia RSA využíva exponent $e = 3$, útočník môže byť schopný zo šifrového textu rekonštruovať príslušnú správu m. V tomto prípade dochádza k úplnej kompromitácii bezpečnosti RSA-OAEP nezávisle na prístupe útočníka k šifrovacím či dešifrovacím orákulám. Ohrozená je teda i neinteraktívna forma komunikácie, pri ktorej je odoslaný iba jediný šifrový text a následne sa už pomocou RSA-OAEP nekomunikuje (napr. prenos symetrických šifrovacích kľúčov). Pri takomto použití RSA-OAEP by napríklad strata bezpečnosti voči útoku s voľbou šifrového textu nemusela nutne znamenať kompromitáciu komunikácie (útočník by nemal k dispozícii dešifrovacie orákulum). Ukážeme si, o čo sa jedná.

Nech teda m označuje správu, ktorú pomocou RSA-OAEP šifrujeme. Pripomeňme si bitové dĺžky jednotlivých vstupov do RSA-OAEP:

$$\text{správa } |m| = n,$$
$$\text{nulové bity } |0^{k_1}| = k_1,$$
$$\text{náhodná hodnota } |r| = k_0.$$

Predpokladajme, že útočník dokáže nejakým spôsobom kompromitovať generátor náhodných čísel, ktorý generuje hodnotu r, tak, že r je z pohľadu útočníka známou

hodnotou (konštantou). V tomto prípade je pre útočníka konštantou i hodnota $G(r)$, pretože v praxi je popis hashovacej funkcie G verejne známy. Označme najvyšších n bitov hodnoty $G(r)$ ako $G(r)^{\{n\}}$, teda:

$$G(r)^{\{n\}} = G(r)[n + k_1 - 1 \dots k_1].$$

Ďalej nech $G(r)_{\{k_1\}}$ značí najnižších k_1 bitov hodnoty $G(r)$, teda:

$$G(r)_{\{k_1\}} = G(r)[k_1 - 1 \dots 0].$$

Analogicky definujeme aj hodnoty $s^{\{n\}}$ a $s_{\{k_1\}}$. Pri kódovaní správy m podľa schémy OAEP je teda vzťah $s = (m||0^{k_1}) \oplus G(r)$ možné prepísať nasledovne:

$$s^{\{n\}} = m \oplus G(r)^{\{n\}},$$

$$s_{\{k_1\}} = 0^{k_1} \oplus G(r)_{\{k_1\}},$$

pričom podľa nášho predpokladu je hodnota $s_{\{k_1\}}$ pre útočníka konštantou. Výsledný šifrový text je teda definovaný vzťahom:

$$y = f(s^{\{n\}}||s_{\{k_1\}}||t),$$

kde f označuje šifrovaciu transformáciu OAEP. V nami uvažovanom prípade RSA-OAEP s exponentom $e = 3$ dostávame:

$$y = (s^{\{n\}}||s_{\{k_1\}}||t)^3 \mod n.$$

Ak si uvedomíme, že násobenie l-tou mocninou dvojky pridáva k bitovému zápisu ľubovoľnej hodnoty l najnižších nulových bitov, môžeme uvedenú rovnosť prepísať nasledovne:

$$y = (s^{\{n\}} \cdot 2^{k_1+k_0} + s_{\{k_1\}} \cdot 2^{k_0} + t)^3 \mod n.$$

Označme hodnotu $s_{\{k_1\}} \cdot 2^{k_0}$ ako κ. Podľa nášho predpokladu je hodnota κ pre útočníka konštantou. Výraz

$$y = (s^{\{n\}} \cdot 2^{k_1+k_0} + \kappa + t)^3 \mod n$$

je teda pre útočníka kubickým polynómom dvoch premenných modulo n s neznámymi $s^{\{n\}}$ a t. Ako ukazuje Coppersmithova práca [**8**], riešenie tohoto špecifického polynómu je za istých okolností možné efektívne vypočítať (riešenie polynómu dvoch premenných modulo n vo všeobecnom prípade je v súčasnosti otvoreným problémom). Coppersmithov prístup spočíva v prevedení uvedeného polynómu modulo n na iný polynóm v okruhu celých čísel, ktorý je už riešiteľný pomocou všeobecného algoritmu publikovaného rovnako v práci [**8**] (tento algoritmus pracuje v polynomiálnom čase a je založený na technike redukcie mreží). Coppersmithov prístup je možné použiť, ak korene polynómu y splňujú nasledujúcu nerovnosť:

$$s^{\{n\}} \cdot t < n^{1/3},$$

kde n je RSA modul. Označme bitovú dĺžku modulu n ako k, teda $k = n+k_0+k_1$. Keďže bitová dĺžka $s^{\{n\}}$ je rovná n a bitová dĺžka t je rovná k_0, jednoduchým výpočtom dostávame:

$$n + k_0 < \frac{k}{3} = \frac{n + k_0 + k_1}{3},$$

$$n + k_0 < \frac{k_1}{2}.$$

Bitová dĺžka		
Modul	Hash	Správa
1024	160	160
2048	224	440
3072	256	750
8192	384	2300
15360	512	4500

Obr. 3.1. Zraniteľné kombinácie RSA-OAEP s exponentom $e = 3$

To znamená, že pokiaľ bitová dĺžka správy m spoločne s bitovou dĺžkou náhodnej hodnoty r tvorí nanajvýš 1/3 celkovej bitovej dĺžky otvoreného textu (resp. modulu), útočník uspeje. Uvažovaná zraniteľnosť je teda aktuálna najmä pre krátke správy m, čiže napríklad vo vyššie spomínanej situácii, kedy je správou symetrický šifrový kľúč. Obrázok 3.1 ilustruje niektoré možné prípady (poznamenajme, že bitová dĺžka hodnoty t je zhodná s bitovou dĺžkou výstupu hashovacej funkcie H, preto je na Obrázku 3.1 reprezentovaná stĺpcom s označením "Hash").
Ako sme si ukázali, aj zdanlivý detail, akým je použitie nekvalitného generátora náhodných čísel, môže i navzdory existujúcemu dôkazu bezpečnosti RSA-OAEP v praxi tento kryptosystém kompromitovať. Rozoberme si ešte v stručnosti, ako je možné sa voči uvedenému útoku brániť:

- používať spoľahlivé generátory náhodných čísel s vysokou mierou entropie a výpočtovou nedosažiteľnosťou predikcie výstupu prípadným útočníkom,
- v obmedzených podmienkach (napr. v prípade smart cards), kde predchádzajúci požiadavok nemôže byť splnený, generovať náhodné čísla ako funkciu výstupu generátora náhodných čísel, ktorý je k dispozícii, a samotnej správy (takto sa dosiahne zvýšenie entropie výslednej náhodnej hodnoty),
- šifrovať dlhšie správy, pričom správu nemožno predĺžiť jednoduchým doplnením o konštantu alebo ľahko odvoditeľnú hodnotu (v tomto prípade sa totiž situácia pre prípadného útočníka nemení).

3.3. Ďalšie problémy spojené s implementáciou

Aj napriek existencii teoretického dôkazu a jeho viac či menej aplikovateľným dôsledkom sa môže bezpečnosť kryptosystému RSA-OAEP v praxi ľahko ukázať ako nedostatočná. Dobrým príkladom je útok, ktorý publikoval Manger vo svojej práci [**14**] z roku 2001. Jedná sa o útok s voľbou šifrového textu, pomocou ktorého za istých okolností dokáže útočník veľmi účinným spôsobom k danému šifrovému textu odhaliť príslušnú správu. Je použiteľný proti kryptosystému RSA-OAEP špecifikovanému v štandarde PKCS#1 verzie 2.0. Táto špecifikácia sa oproti klasickej schéme OAEP, s ktorou pracujeme v tomto texte, v podstate odlišuje v tom, že šifrovacia transformácia RSA sa namiesto výrazu $(s||t)$ aplikuje na výraz $(00||s||t)$. Pred umocnením mod n sa teda k získanému výrazu pripojí navyše ešte jeden nulový bajt. Takto je zaistené, aby bola výsledná hodnota nižšia než RSA modul, čo je potrebné pre dosiahnutie injektivity funkcie RSA. Pri dešifrovaní šifrového textu sa tak vo finále vždy aplikujú dva typy kontrol:

- kontrola na nulovosť najvyššieho bajtu po umocnení privátnym kľúčom;

- kontrola na korektnosť štruktúry otvoreného textu (bez najvyššieho bajtu) po dekódovaní podľa schémy OAEP, tj. overenie výskytu určeného počtu nulových bajtov na určených pozíciách.

Označme RSA modul ako n, jeho bajtovú dĺžku ako $k = \lceil \log_{256} n \rceil$ a nakoniec položme $B = 2^{8(k-1)}$. Predpokladajme, že útočník má (okrem verejného kľúča (n, e)) k dispozícii orákulum, ktoré dokáže pre ľubovoľný šifrový text $c = x^e \mod n$ odpovedať, ktorú z relácií $x < B$ alebo $x \geq B$ príslušný otvorený text $x = c^d \mod n$ spĺňuje. Toto orákulum – nazvime ho rozhodovacie – teda pre ľubovoľný šifrový text odpovedá, či príslušný otvorený text prešiel kontrolou na nulovosť najvyššieho bajtu ($x < B$), alebo nie ($x \geq B$). Samotný útok potom prebieha nasledovne. Útočník teda chce za pomoci rozhodovacieho orákula rekonštruovať hodnotu x zo znalosti c. Za týmto účelom bude postupne posielať šifrové texty tvaru $c' = c \cdot f^e \mod n$ rozhodovaciemu orákulu. To mu následne odpovie, ktorá z relácií $x \cdot f < B$ alebo $x \cdot f \geq B$ je splnená. Odpoveď rozhodovacieho orákula pre útočníka vlastne znamená rozhodnutie, či $xf \in [0, B) \mod n$ alebo $xf \in [B, n) \mod n$. Takéto rozhodnutie, nech už dopadne akokoľvek, vylúčením jednej z dvoch možností redukuje iterval, v ktorom sa príslušné x môže nachádzať. Nakoniec sa interval zúži tak, že ostane už iba jediná hodnota, ktorou je hľadaný otvorený text x. Konkrétny spôsob, akým útočník pri voľbe postupnosti hodnôt f postupuje, je možné nájsť v publikácii [**14**].

Akým spôsobom môže útočník v realite získať prístup k niečomu, ako je rozhodovacie orákulum? Pre zodpovedanie tejto otázky si stačí uvedomiť, že rozhodovacie orákulum môže byť sprostredkované prakticky akýmkoľvek spôsobom, pomocou ktorého útočník dokáže rozpoznať, či ním zadaný šifrový text prešiel kontrolou na nulovosť najvyššieho bajtu, alebo nie. Ak svoje dotazy na dešifrovanie útočník posiela nejakému automatu (napr. internetovému serveru prevadzkujúcemu nejakú službu), ktorý pred dešifrovaním pomocou svojho privátneho kľúča najskôr kontroluje nulovosť najvyššieho bajtu, môže mu tento automat za istých okolností zároveň poslúžiť aj ako rozhodovacie orákulum, a to minimálne nasledujúcimi spôsobmi:

- rozdielna syntax chybových správ – akákoľvek odlišnosť medzi správami oznamujúcimi útočníkovi, že zadaný šifrový text neprešiel pri spracovaní jednou z kontrol, je dostatočná. Rozdiel pritom nemusí byť úmyselný, stačí, ak sa správy líšia o jedinú medzeru, bodku či veľké písmeno. Na tento problém štandard PKCS#1 verzie 2.0 explicitne upozorňuje. Konkrétne uvádza, že je dôležité, aby chybové správy oznamujúce nesplnenie niektorej z kontrol boli rovnaké. Napriek tomu však v mnohých implementáciách RSA-OAEP nie je toto doporučenie dodržané.
- ďalšie chybové situácie – jedná sa o implementačné nedostatky, ktoré je útočník schopný využiť k zámernému vyvolaniu nejakej neštandardnej chyby vo fáze dekódovania šifrového textu. Vzhľadom k tomu, že táto fáza nastáva až po úspešnej kontrole na nulovosť najvyššieho bajtu, obdrží útočník v prípade jej úspešného vyvolania potrebnú informáciu. Príkladom uvedeného typu útoku môže byť neplatná špecifikácia hashovacej funkcie, ktorá má byť použitá pri dekódovaní šifrového textu. Možnou obranou je opäť unifikácia všetkých chybových správ z pohľadu útočníka.

- logy – aj keď v danej implementácii RSA-OAEP neexistuje rozdiel medzi chybovými správami odosielanými útočníkovi pri rôznych chybových situáciách, bude tento rozdiel pravdepodobne existovať v popise jednotlivých chybových situácií v systémových auditných záznamoch (logoch). Je tomu tak preto, aby vývojári, podpora a samotní užívatelia mali na základe spätnej väzby možnosť systém monitorovať, lepšie mu porozumieť či v prípade potreby adekvátne reagovať. Tieto logy sú väčšinou prístupné širšiemu spektru ľudí než privátny kľúč, čo zväčšuje útočníkove možnosti. Preto je potrebné prístup k systémovým logom riadiť – udeľovať/odoberať ho iba na základe stanovených pravidiel a priebežne ho monitorovať.
- časovanie – identické chovanie systému pri komunikácii s útočníkom (jednotné chybové hlášky) ani znemožnenie prístupu k logom ešte stále spoľahlivo nechráni systém pred útokom s využitím rozhodovacieho orákula. Dôvodom je, že detekcia rozličných chýb môže trvať rozličné množstvo času. Ak je tento časový rozdiel merateľný, predstavuje pre útočníka možnosť využitia ako rozhodovacie orákulum. V prípade RSA-OAEP existuje dokonca možnosť tento rozdiel ovplyvňovať. Pri dekódovaní sa totiž využíva hash parametrov, ktoré automatu zadáva útočník spolu so šifrovým textom. Tento hash sa podľa doporučenia štandardu PKCS#1 verzie 2.0 vo väčšine prípadov počíta bezprostredne pred nutnosťou použitia – a teda až po kontrole na nulovosť najvyššieho bajtu. Útočníkovi sa tým ponúka možnosť zaslať ako parametre pre výpočet hashu dostatočne veľký objem dát (kľudne i niekoľko megabajtov) tak, aby bol schopný rozpoznať rozdiel. Týmto opäť dostáva potrebné rozhodovacie orákulum. Efektívnou obranou je výpočet spomínaného hashu ešte pred prvou kontrolou na nulovosť najvyššieho bajtu.

Uvedeným nedostatkom sa aktuálny štandard PKCS#1 verzie 2.1 už snaží predchádzať. Explicitne sa v ňom uvádza, že jednotlivé chyby musia byť z pohľadu útočníka nerozlíšiteľné a čas behu nesmie indikovať, ktorá z možných chýb nastala.

Popísaný Mangerov útok voči PKCS#1 verzie 2.0 je svojim charakterom veľmi podobný Bleichenbacherovmu útoku voči PKCS #1 verzie 1.5, ktorým sme sa zaoberali na začiatku kapitoly. Je ale omnoho efektívnejší, pretože jeho časová zložitosť je logaritmická v počte dotazov na rozhodovacie orákulum. Pre porovnanie – v prípade 1024-bitovej veľkosti modulu n vyžaduje tento útok iba približne 1100 dotazov na rozhodovacie orákulum, zatiaľ čo Bleichenbacherov útok vyžaduje až približne tisícnásobne viac dotazov. Hlavný rozdiel medzi oboma útokmi však spočíva v tom, že Bleichenbacher útočí na samotnú schému (a bol eliminovaný jej zmenou), pričom Manger útočí na jej konkrétnu implementáciu (a principiálne sa eliminovať nedá). Existencia Mangerovho útoku by mala slúžiť ako memento, že aj napriek korektnému dôkazu bezpečnosti môže byť v praxi veľmi jednoduché záruky z toho plynúce vynulovať nedokonalou implementáciou.

Možným spôsobom obrany voči útokom podobného typu je výmena poradia, v akom budú prebiehať kontroly pri spracovaní šifrového textu od útočníka. Šifrový text je tak po umocnení privátnym kľúčom najskôr podrobený testu na korektnosť štruktúry, pričom hodnota najvyššieho bajtu je zatiaľ úplne ignorovaná. Následne prichádza dekódovanie podľa schémy OAEP a overenie výskytu určeného počtu nulových bajtov na určených pozíciách. V prípade kryptosystému RSA-OAEP

popísaného v úvode tejto kapitoly musí teda pre splnenie kontroly na konci dekódovania ostať k_1 núl medzi správou dĺžky n a náhodným semienkom dĺžky k_0. Pravdepodobnosť, že sa tak stane pre šifrový text tvaru $c' = c \cdot f^e \mod n$ z predchádzajúceho odstavca, je rovná výrazu $1/2^{k_1}$, a teda zanedbateľná. Dôvodom je, že útočník nedokáže stanoviť hodnoty f tak, aby príslušný súčin $c' = c \cdot f^e \mod n$ obsahoval k_1 nulových bajtov na požadovanej pozícii, pretože funkcia RSA (mocnenie mod n) je jendosmerná. Takže ak bude útočník postupovať podľa Mangerovho návodu, dostane sa k druhej kontrole na nulovosť najvyššieho bajtu – a tým k možnosti jednotlivé kontroly rozlíšiť a získať tak rozhodovacie orákulum – iba v zanedbateľnom (prakticky nulovom) zlomku prípadov.

4
Bezpečnosť RSA-OAEP a štandardný model

Táto kapitola je venovaná výsledkom preukázateľnej bezpečnosti RSA-OAEP v štandardnom modele. Vychádzať budeme z dvojice povšimnutia hodných článkov z nedávnej minulosti. Zámerom kapitoly je ukázať, kam v súčasnosti siahajú možnosti štandardného modelu pre dokazovanie bezpečnosti RSA-OAEP.

4.1. Dokázateľná nedokázateľnosť bezpečnosti RSA-OAEP voči CCA2

Východzím podkladom pre tento oddiel je práca [**12**] dvojice Kiltz a Pietrzak z roku 2009. Autori tu ukázali, že v štandardnom modele nie je možné dokázať bezpečnosť žiadneho kryptosystému s verejným kľúčom a náhodným dopĺňaním správy – špeciálne teda RSA-OAEP – ani na základe veľmi silného predpokladu, akým je existencia ideálnych permutácii s padacími vrátkami.

4.1.1. Ideálna permutácia s padacími vrátkami

Začneme formalizáciou základných pojmov. Písmenom k budeme ďalej označovať bezpečnostný parameter.

Poznámka. Skutočnosť, že algoritmus A má na vstupe hodnotu x, počas svojho behu má prístup k algoritmu O a jeho výstupom je hodnota z, budeme označovať ako

$$A^O(x) = z.$$

Definícia (permutácia s padacími vrátkami). Hovoríme, že usporiadaná trojica pravdepodobnostných polynomiálnych algoritmov $(\mathrm{Tdg}, \mathrm{F}, \mathrm{F}^{-1})$ implementuje *permutáciu s padacími vrátkami*, ak platí:

- $\mathrm{Tdg}(1^k) = (ek, td)$,
- $\mathrm{F}(ek, \cdot)$ implementuje nejakú permutáciu $f_{ek}(\cdot)$ na $\{0,1\}^k$,
- $\mathrm{F}^{-1}(td, \cdot)$ implementuje jej inverziu $f_{ek}^{-1}(\cdot)$.

Nech A, G predstavujú dva vzájomne interagujúce pravdepodobnostné polynomiálne algoritmy, pričom algoritmus A nazveme útočníkom a algoritmus G nazveme hrou. Ďalej nech τ označuje nejakú permutáciu na na $\{0,1\}^k$. Výsledkom (výstupom) hry G nad τ je bit $d \in \{0,1\}$. Jeden beh hry G s útočníkom A a výsledkom d budeme zapisovať ako $\mathrm{Exp}_\tau^G(A) = d$.

Definícia. Nech $\mathcal{P}_k$ označuje množinu všetkých permutácií na $\{0,1\}^k$ a nech $\tau \in \mathcal{P}_k$ označuje nejakú náhodnú permutáciu z $\mathcal{P}_k$. Hovoríme, že útočník A *vyhráva* hru G nad τ, ak $d = \mathrm{Exp}_\tau^G(A) = 1$. Jeho *výhodu* v hre G nad τ budeme

označovať ako $\mathrm{Adv}_{\tau}^{G}(A,k)$, teda

$$\mathrm{Adv}_{\tau}^{G}(A,k) = \Pr[d = 1 : \mathrm{Exp}^{G^{\tau(\cdot)}}(A^{\tau(\cdot)}(1^k)) = d].$$

Hra G nad τ sa pre hodnotu $0 \leq \delta \leq 1$ nazýva *δ-ťažká*, ak pre každého pravdepodobnostného polynomiálneho útočníka A platí

$$\mathrm{Adv}_{\tau}^{G}(A,k) \approx \delta,$$

teda ak je funkcia $|\mathrm{Adv}_{\tau}^{G}(A,k) - \delta|$ zanedbateľná v k. *Náročnosťou* hry G nad τ budeme rozumieť najmenšie δ také, že G je δ-ťažká pre nejakú náhodnú permutáciu z $\mathcal{P}_k$ Túto hodnotu budeme označovať ako $\delta(G)$, je teda

$$\delta(G) = \min_{\tau \in \mathcal{P}_k}\{0 \leq \delta \leq 1 : \mathrm{Adv}_{\tau}^{G}(A,k) \approx \delta\}.$$

Predchádzajúca definícia umožňuje formálne modelovať rôzne bezpečnostné vlastnosti náhodnej permutácie $\tau(\cdot)$ vzhľadom k útočníkovi A. Napríklad jednosmernosť náhodnej permutácie τ na $\{0,1\}^k$ je možné popísať nasledovne. Hra G zvolí náhodne a uniformne $x \in \{0,1\}^k$, spočíta $y = \tau(x)$ a ako výsledok vráti $d = 1$, ak platí:

$$(A^{\tau(\cdot)}(y) = x') \ \&\ (x' = x).$$

Náhodná permutácia $\tau(\cdot)$ je potom jednosmerná, ak je výhoda ľubovoľného pravdepodobnostného polynomiálneho útočníka

$$\mathrm{Adv}_{\tau}^{G}(A,k) \leq \frac{q_A^{\tau} + 1}{2^k},$$

kde výraz q_A^{τ} označuje počet dotazov útočníka A na permutáciu $\tau(\cdot)$. Náročnosť hry G je v tomto prípade $\delta(G) = 0$.

Definícia (ideálna permutácia s padacími vrátkami). Nech $\mathrm{TDP} = (\mathrm{Tdg}, \mathrm{F}, \mathrm{F}^{-1})$ je permutácia s padacími vrátkami. Výhoda útočníka A v hre G je teda charakterizovaná výrazom

$$\mathrm{Adv}_{\mathrm{TDP}}^{G}(A,k) = \Pr[d = 1 : \mathrm{Tdg}(1^k) = (ek, td);\ \mathrm{Exp}^{G^{\mathrm{F}(ek,\cdot)}}(A^{\mathrm{F}(ek,\cdot)}(1^k)) = d].$$

Hovoríme, že permutácia s padacími vrátkami TDP je *bezpečná* vzhľadom k hre G, ak pre každého pravdepodobnostného polynomiálneho nepriateľa A platí $\mathrm{Adv}_{\mathrm{TDP}}^{G}(A,k) \approx \delta(G)$. Nakoniec, TDP nazývame *ideálnou* permutáciou s padacími vrátkami, ak je bezpečná pre každú pravdepodobnostnú polynomiálnu hru G.

Ideálna permutácia s padacími vrátkami je veľmi silnou konštrukciou, pretože podľa uvedenej definície by musela vykazovať úplne všetky bezpečnostné vlastnosti zároveň. Jedná sa teda iba o teoretický koncept a v skutočnosti žiadna ideálna permutácia s padacími vrátkami neexistuje. Cieľom je však dospieť k nemožnosti dokázať bezpečnosť ľubovoľného kryptosystému s verejným kľúčom a náhodným doplňaním správy. K tomuto cieľu dospejeme tak, že ukážeme neexistenciu kryptosystému s verejným kľúčom a náhodným doplňaním správy, ktorého bezpečnosť voči adaptívnemu útoku s voľbou šifrového textu by bolo možné redukovať na bezpečnosť ideálnej permutácie s padacími vrátkami. Tým skôr nemôže existovať ani kryptosystém s verejným kľúčom a náhodným doplňaním správy, ktorého CCA2 bezpečnosť by bolo možné redukovať na bezpečnosť ľubovoľnej permutácie s padacími vrátkami, ktorá je bezpečná len voči nejakej realistickej podmnožine ťažkých hier.

4.1.2. Schéma náhodného dopĺňania správy

Definícia (Schéma náhodného dopĺňania správy). Nech μ, ρ sú prirodzené čísla splňujúce $\mu + \rho \leq k$. *Schéma náhodného dopĺňania správy* $\Pi = (\pi, \hat{\pi})$ je dvojicou zobrazení

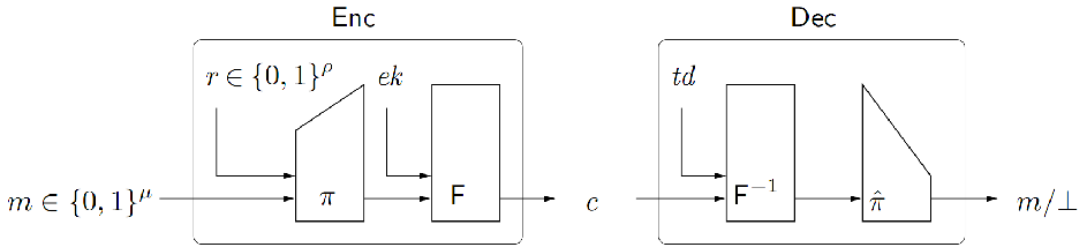
$$\pi : \{0,1\}^{\mu} \times \{0,1\}^{\rho} \to \{0,1\}^{k},$$
$$\hat{\pi} : \{0,1\}^{k} \to \{0,1\}^{\mu} \cup \{\perp\},$$

kde π je injektívne a navyše platí nasledujúca rovnosť:

$$(\forall m \in \{0,1\}^{\mu}, r \in \{0,1\}^{\rho}) : \hat{\pi}(\pi(m||r)) = m.$$

Definícia (Kryptosystém s náhodným dopĺňaním správy a jednosmernou permutáciou s padacími vrátkami). Nech $\mu, \rho : \mathbb{N} \to \mathbb{N}$ sú funkcie splňujúce $(\forall k \in \mathbb{N}) : (\mu(k) + \rho(k) \leq k)$. Trojicu algoritmov $\mathrm{PKE} = (\mathrm{Kg}, \mathrm{Enc}, \mathrm{Dec})$ polynomiálnych v k nazývame *kryptosystém s náhodným dopĺňaním správy a jednosmernou permutáciou s padacími vrátkami* s priestorom správ μ a priestorom náhodnosti ρ, ak pre každú jednosmernú permutáciu s padacími vrátkami $\mathrm{TDP} = (\mathrm{Tdg}, \mathrm{F}, \mathrm{F}^{-1})$ platí:

- algoritmus $\mathrm{Kg}(k)$ vráti verejný kľúč $pk = (ek, \Pi)$ a privátny kľúč $sk = td$, pričom $\mathrm{Tdg}(1^k) = (ek, td)$ a $\Pi = (\pi, \hat{\pi})$ je dvojicou jednosmerných permutácií s padacími vrátkami vyhovujúcich predchádzajúcej definícii.
- algoritmus $\mathrm{Enc}(pk, m)$ ku správe $m \in \{0,1\}^{\mu}$ vráti šifrový text $c = f_{ek}(\pi(m||r))$, pričom hodnotu $r \in \{0,1\}^{\rho}$ zvolí z danej množiny náhodne a uniformne.
- algoritmus $\mathrm{Dec}(sk, c)$ zo šifrového textu $c \in \{0,1\}^{k}$ vráti hodnotu $m = \hat{\pi}(f_{ek}^{-1}(c)) \in \{0,1\}^{\mu} \cup \{\perp\}$.



Obr. 4.1. Schéma kryptosystému s náhodným dopĺňaním správy $\Pi = (\pi, \hat{\pi})$ a jednosmernou permutáciou s padacími vrátkami $\mathrm{TDP} = (\mathrm{Tdg}, \mathrm{F}, \mathrm{F}^{-1})$

Uvedenú definíciu graficky ilustruje Obrázok 4.1.2. Všimnime si, že táto definícia presne nešpecifikuje, ako sa algoritmus $\mathrm{Dec}(pk, \cdot)$ zachová v prípade, že na vstupe dostane neplatný šifrový text. Dôležitým je tiež detail, že $\hat{\pi}$ je ako komponenta $\Pi = (\pi, \hat{\pi})$ súčasťou verejného kľúča. Takto zabránime, aby schéma s náhodným dopĺňaním správy Π bola sama o sebe kryptosystémom bezpečným voči útoku s voľbou šifrového textu (podrobnejšie rozoberieme v nasledujúcom oddieli).

Kryptosystém s náhodným dopĺňaním správy a jednosmernou permutáciou s padacími vrátkami je obecným pojmom, ktorému okrem iného vyhovuje aj OAEP (viď oddiel 2.2.1) a jeho rôzne varianty – či už je to Shoupov OAEP+ (článok [**19**]) alebo Bonehov SAEP a SAEP+ (článok [**5**]).

4.1.3. Nemožnosť dokázania bezpečnosti RSA-OAEP voči CCA2

Nasleduje hlavné tvrdenie tejto kapitoly. Toto tvrdenie hovorí, že neexistuje kryptosystém PKE = (Kg, Enc, Dec) s náhodným dopĺňaním správy taký, že ľubovoľný útočník schopný proti PKE úspešne realizovať útok s voľbou šifrového textu by mohol byť využitý k zlomeniu bezpečnosti príslušnej TDP ako ideálnej permutácie s padacími vrátkami.

TVRDENIE 4.1. *Neexistuje redukcia bezpečnosti ľubovoľného kryptosystému s náhodným dopĺňaním správy proti adaptívnemu útoku s voľbou šifrového textu (CCA2) na bezpečnosť ideálnej permutácie s padacími vrátkami.*

DÔKAZ. Bez újmy na všeobecnosti zvoľme nejaký kryptosystém s náhodným dopĺňaním správy PKE = (Kg, Enc, Dec) s priestorom správ $\mu(k)$ a priestorom náhodnosti $\rho(k)$. Predpokladajme, že neplatí $\mu(k) + \rho(k) \in \mathcal{O}(\log(k))$ (v opačnom prípade by totiž bezpečnosť každého PKE bola triviálne zlomiteľná útokom hrubou silou).
Pokračovať budeme definovaním dvoch orákul T a B, na ktorých konštrukcii je tento dôkaz založený.

DEFINÍCIA (orákulum T implementujúce permutáciu s padacími vrátkami). Nech $\mathcal{P}_k$ označuje množinu všetkých permutácií na $\{0,1\}^k$. Pre každé $k \in \mathbb{N}$ zvolíme z $\mathcal{P}_k$ náhodne a uniformne $2^k + 1$ permutácií $f_{k,0}, \ldots, f_{k,2^k-1}$ a g_k. Orákulum $T = (T_1, T_2, T_3)$ je trojicou pravdepodobnostných polynomiálnych algoritmov splňujúcich nasledujúce predpisy:

- konverzia padacích vrátok na verejný kľúč: $T_1(td) = g_k(td) = ek$, kde $k = |td|$.
- vyhodnotenie: $T_2(ek, x) = f_{k,ek}(x)$, kde $|ek| = |x| = k$.
- inverzia: $T_3(td, y) = f^{-1}_{k,g_k(td)}(y)$, kde $|td| = |y| = k$.

Orákulum T je iba prepisom defnície permutácie s padacími vrátkami.

DEFINÍCIA (orákulum B použité k zlomeniu kryptosystému). Orákulum B je pravdepodobnostným polynomiálnym algoritmom, ktorý prijíma dva typy vstupov a vracia k nim dva typy výstupov:

- na vstupe (k, ek, Π), kde $k \in \mathbb{N}$ je bezpečnostný parameter, $ek \in \{0,1\}^k$ je šifrovací kľúč a $\Pi = (\pi, \hat{\pi})$ je dvojicou zobrazení $\pi : \{0,1\}^{\mu(k)+\rho(k)} \to \{0,1\}^k$ a $\hat{\pi} : \{0,1\}^k \to \{0,1\}^{\mu(k)}$ (Π je schéma náhodného dopĺňania správy) orákulum B vráti ako svoj výstup vektor šifrových textov $[c_1, \ldots, c_{4k}]$, pričom
$$c_i = f_{k,ek}(\pi(m_i || r_i)),\ i = 1, \ldots, 4k,$$
kde hodnoty $m_i \in \{0,1\}^{\mu(k)}$ a $r_i \in \{0,1\}^{\rho(k)}$ sú volené náhodne a uniformne. Ak je teda dvojica (ek, π) verejným kľúčom nejakého kryptosystému s náhodným dopĺňaním správy, potom c_i sú korektné šifrové texty tohto kryptosystému.
- na vstupe $(k, ek, \pi, [m'_1, \ldots, m'_{4k}])$ orákulum B overí, či platí $[m'_1, \ldots, m'_{4k}] = [m_1, \ldots, m_{4k}]$, kde m_i pre $i = 1, \ldots, 4k$ sú správy, ktoré by B volilo na vstupe (k, ek, π) z predchádzajúcej odrážky. Ak sa uvedené vektory zhodujú, B vráti ako svoj výstup dešifrovací kľúč $td = g_k^{-1}(ek)$ odpovedajúci šifrovaciemu kľúču ek, ktorý B obdržalo na vstupe. V opačnom prípade vráti orákulum B ako svoj výstup symbol $\perp$.

Všimnime si, že orákulum B v skutočnosti predstavuje algoritmus redukcie. Podľa svojho popisu totiž B najprv vygeneruje náhodné šifrové texty, ktoré vráti útočníkovi. Ak následne v druhom kroku obdrží B od útočníka otvorené texty odpovedajúce náhodným šifrovým textom z prvého kroku, B je vďaka tomu schopné (v polynomiálnom čase) vrátiť ako svoj výstup padacie vrátka td. Inak povedané – útočníka, ktorý dokáže zlomiť uvažovaný kryptosystém, je orákulum B reprezentujúce algoritmus redukcie schopné využiť k vyriešeniu príslušného nedosažiteľného matematického problému (invertovaniu TDP).
Aby sme dokázali tvrdenie Vety, stačí nám teda overiť, že vyššie definované orákulá T a B splňujú nasledujúce body:

1) existuje permutácia s padacími vrátkami $\mathrm{TDP} = (\mathrm{Tdg}, \mathrm{F}, \mathrm{F}^{-1})$, ktorú orákulum T implementuje (toto budeme označovať zápisom TDP^T).
2) TDP^T je ideálna permutácia s padacími vrátkami a to aj v prípade, že má útočník A v každej hre prístup k orákulám T a B (toto budeme označovať zápisom $A^{T,B}$).
3) ľubovoľný kryptosystém PKE s náhodným dopĺňaním správy, v ktorom sa ako permutácia s padacími vrátkami využíva TDP^T (toto budeme označovať zápisom $\mathrm{PKE}^{\mathrm{TDP}^T}$), nie je bezpečný voči adaptívnemu útoku s voľbou šifrového textu relizovanom útočníkom $A^{T,B}$.

Uvedené podmienky overíme pomocou niekoľkých nasledujúcich Lemm.

Lemma 4.2. *Existuje pravdepodobnostná polynomiálna permutácia s padacími vrátkami TDP taká, že TDP^T je implementáciou permutácie s padacími vrátkami.*

Dôkaz. Implementáciu $\mathrm{TDP}^T = (\mathrm{Tdg}, \mathrm{F}, \mathrm{F}^{-1})$ skonštruujeme nasledovne:

- $\mathrm{Tdg}(1^k)$ najskôr náhodne a uniformne zvolí $td \in \{0,1\}^k$, následne spočíta odpovedajúcu hodnotu $ek = T_1(td)$ a ako výstup vráti dvojicu (ek, td).
- $\mathrm{F}(ek, x)$ ako výstup vráti $T_2(ek, x)$.
- $\mathrm{F}^{-1}(td, y)$ ako výstup vráti $T_3(td, y)$.

Podľa definície permutácie s padacími vrátkami sme s dôkazom tejto Lemmy hotoví. Tým sme splnili vyššie uvedený bod 1).

□

Lemma 4.3. TDP^T *je ideálnou permutáciou s padacími vrátkami s pravdepodobnosťou 1 cez všetky voľby orákula T.*

Dôkaz. $\mathrm{TDP}^T = (\mathrm{Tdg}, \mathrm{F}, \mathrm{F}^{-1})$ ako v dôkaze predchádzajúcej Lemmy. Uvažujme ľubovoľnú δ-ťažkú hru G. Výhoda útočníka A v tejto hre je daná výrazom

$$\mathrm{Adv}^G_{\mathrm{TDP}^T}(A,k) = \Pr[d = 1 : \mathrm{Tdg}(1^k) = (ek, td);\ \mathrm{Exp}^{G^{\mathrm{F}(ek,\cdot)}}(A^{\mathrm{F}(ek,\cdot)}(ek)) = d].$$

Ďalej nech T_{ek} označuje T, kde permutáciu $f_{k,ek}(\cdot)$ nahradíme novou náhodne a uniformne zvolenou permutáciou $\tau(\cdot)$ na $\{0,1\}^k$ (toto budeme označovať ako $\tau \leftarrow_{rnd} \mathcal{P}_k$). Odpovedajúci výraz $\mathrm{Adv}^G_{\mathrm{TDP}^{T_{ek}}}(A,k)$ je potom rovný

$$\Pr[d = 1 : \mathrm{Tdg}(1^k) = (ek, td);\ \tau(\cdot) \leftarrow_{rnd} \mathcal{P}_k;\ \mathrm{Exp}^{G^{\tau(\cdot)}}(A^{\tau(\cdot)}(ek)) = d].$$

Podľa definície δ-ťažkej hry platí $\mathrm{Adv}^G_{\mathrm{TDP}^{T_{ek}}}(A,k) \approx \delta$. Potrebujeme ukázať aj platnosť vzťahu

$$\mathrm{Adv}^G_{\mathrm{TDP}^T}(A,k) \approx \mathrm{Adv}^G_{\mathrm{TDP}^{T_{ek}}}(A,k).$$

Tento vzťah hovorí, že žiaden pravdepodobnostný polynomiálny útočník A nie je schopný rozlíšiť medzi jednotlivými hrami s permutáciami $f_{k,ek}$ a τ s pravdepodobnosťou, ktorá nie je zanedbateľná. Podľa práce [**10**] je náhodná permutácia jednosmerná skoro iste (tzn. limita pravdepodobnosti tohoto javu pre $k \to \infty$ je rovná 1), preto je útočník A schopný k danej hodnote $ek = T_1(td)$ získať príslušnú hodnotu td iba so zanedbateľnou pravdepodobnosťou. Ďalej permutácia $f_{k,ek}$ je volená z exponenciálne veľkej množiny (indexovanej prvkami $\{0,1\}^k$), takže môže byť pravdepodobnostným polynomiálnym útočníkom A od náhodne volenej permutácie τ odlíšená iba so zanedbateľnou výhodou. Vďaka tomu sú distribúcie výsledkov d jednotlivých hier s permutáciami $f_{k,ek}$ a τ z pohľadu útočníka A nerozlíšiteľné. To znamená, že $|\mathrm{Adv}^G_{\mathrm{TDP}^T}(A,k) - \mathrm{Adv}^G_{\mathrm{TDP}^{T_{ek}}}(A,k)|$ je funkcia zanedbateľná v k a teda platí $\mathrm{Adv}^G_{\mathrm{TDP}^T}(A,k) \approx \mathrm{Adv}^G_{\mathrm{TDP}^{T_{ek}}}(A,k)$, ako sme chceli ukázať. Relácia $\approx$ je ekvivalenciou a využitím jej tranzitivity dostávame

$$(\mathrm{Adv}^G_{\mathrm{TDP}^{T_{ek}}}(A,k) \approx \delta) \;\&\; (\mathrm{Adv}^G_{\mathrm{TDP}^T}(A,k) \approx \mathrm{Adv}^G_{\mathrm{TDP}^{T_{ek}}}(A,k))$$
$$\Downarrow$$
$$\mathrm{Adv}^G_{\mathrm{TDP}^T}(A,k) \approx \delta.$$

Nakoniec, keďže sme tento vzťah odvodili pre ľubovoľnú δ-ťažkú hru G, konštrukcia TDP^T je podľa definície ideálnou permutáciou s padacími vrátkami.

□

Lemma 4.4. *Existuje pravdepodobnostný polynomiálny útočník A, ktorý s prístupom k orákulám T a B (útočník $A^{T,B}$) úspešne realizuje adaptívny útok s voľbou šifrového textu voči ľubovoľnému kryptosystému* PKE *s náhodným dopĺňaním správy, kde sa ako permutácia s padacími vrátkami využíva* TDP^T *(kryptosystém* PKE^{TDP^T}*).*

Dôkaz. Útočník A pri adaptívnom útoku s voľbou šifrového textu najskôr obdrží verejný kľúč (ek, Π). Následne položí dotaz orákulu B v tvare (k, ek, Π) a ako odpoveď dostane vektor šifrových textov $[c_1, \ldots, c_{4k}]$. V ďalšom kroku sa dotáže dešifrovacieho orákula (toto orákulum má útočník k dispozícii podľa definície scenára adaptívneho útoku s voľbou šifrového textu, viď oddiel 1.2.2) práve vektorom $[c_1, \ldots, c_{4k}]$, pričom ako odpoveď obdrží vektor príslušných otvorených textov $[m'_1, \ldots, m'_{4k}]$. Nakoniec, útočník A sa dotáže orákula B vstupom $(k, ek, \Pi, [m'_1, \ldots, m'_{4k}])$. Keďže rovnosť $[m'_1, \ldots, m'_{4k}] = [m_1, \ldots, m_{4k}]$ bude podľa definície orákula B splnená, obdrží A ako odpoveď dešifrovací kľúč td. So znalosťou tohto dešifrovacieho kľúča je útočník schopný cieľový šifrový text c^* priamo dešifrovať (dokáže invertovať príslušnú permutáciu s padacími vrátkami), čo vedie k úspešnej realizácii adaptívneho útoku s voľbou šifrového textu podľa tvrdenia Lemmy.

□

Práve dokázanou Lemmou sme splnili vyššie uvedený bod 3). K dokončeniu dôkazu Tvrdenia 4.1 nám ostáva ukázať platnosť bodu 2), ktorý je zovšeobecnenou verziou Lemmy 4.3. Dosiahnuť tento cieľ nám technicky umožní nasledujúca Lemma.

Lemma 4.5. *Nech* $\mathrm{TDP}^T = (\mathrm{Tdg}, \mathrm{F}, \mathrm{F}^{-1})$*, kde* $\mathrm{Tdg}(1^k) = (ek, td)$*. Uvažujme ľubovoľný pravdepodobnostný polynomiálny algoritmus C, ktorému je umožnený prístup k orákulám T a B. Nech $C^{T,B}(ek)$ realizuje pri svojom výpočte najviac $q(k)$*

dotazov, kde $q(k) \leq 2^{\mu(k)+\rho(k)}/2$. *Potom existuje pravdepodobnostný polynomiálny simulátor* S *taký, že výstupy* $C^{T,B}(ek)$ *a* $C^{T,S^T}(ek)$ *sú výpočtovo nerozlíšiteľné (ich rozlíšenie ako dvoch štatistických súborov je výpočtovo nedosažiteľná úloha), pričom predpokladáme, že simulátor* S *má prístup práve k dotazom* C *na* T, *ktoré využívajú* $\mathrm{F}^{-1}(\cdot,\cdot)$ *(pri dotazoch na dešifrovanie teda musí* C *okrem šifrového textu posielať* T *navyše i dešifrovací kľúč).*

Dôkaz. Simulátor S definujeme nasledovne:

- Na vstupe (k, ek, Π) zvolí S náhodne a uniformne hodnoty $m_i \in \{0,1\}^{\mu(k)}$ a $r_i \in \{0,1\}^{\rho(k)}$ pre $i = 1, \ldots, 4k$. Následne spočíta $c_i = f_{k,ek}(\pi(m_i||r_i))$ a ako svoj výstup vráti vektor šifrových textov $[c_1, \ldots, c_{4k}]$. Záznam $(k, ek, \Pi, [c_1, \ldots, c_{4k}])$ si simulátor S uloží do zoznamu, aby v prípade, že na vstupe obdrží opäť rovnakú trojicu (k, ek, Π), vrátil ako svoj výstup vždy rovnaký vektor šifrových textov $[c_1, \ldots, c_{4k}]$. Tým dosiahne konzistenciu svojich odpovedí na dotazy. Všimnime si, že S počíta svoj výstup rovnako, ako orákulum B podľa svojej definície.
- Na vstupe $(k, ek, \pi, [m'_1, \ldots, m'_{4k}])$ najprv S overí, či v jeho zozname existuje k trojici (k, ek, π) odpovedajúci záznam.
  - Ak príslušný záznam neexistuje, vráti S ako svoj výstup $\perp$, čím presne simuluje orákulum B až na zanedbateľné množstvo prípadov, kedy hodnoty m'_i náhodou odpovedajú hodnotám m_i, ktoré by B použilo na vstupe (k, ek, π). Pravdepodobnosť tejto udalosti je $2^{-4k\mu(k)}$.
  - Ak príslušný záznam existuje, postupuje S nasledovne:
    - Ak C položilo S dotaz (k, ek, Π), označme $[m_1, \ldots, m_{4k}]$ vektor správ, ktorým by S na tento dotaz C normálne odpovedal. V prípade, že $[m_1, \ldots, m_{4k}] \neq [m'_1, \ldots, m'_{4k}]$, vráti S ako svoj výstup $\perp$, čím presne simuluje orákulum B.
    - V opačnom prípade, teda ak $[m_1, \ldots, m_{4k}] = [m'_1, \ldots, m'_{4k}]$, overí S, či pre všetky dotazy C na T tvaru (td, x), ktoré využívajú F^{-1}, platí rovnosť $ek = \mathrm{Tdg}(td)$. Ak táto rovnosť platí, vráti S ako dopoveď na dotaz dešifrovací kľúč td, presne ako by to urobilo orákulum B (hodnotu td simulátor S pozná vďaka prístupu ku všetkým dotazom C na T podľa predpokladu tvrdenia Lemmy). Jediným prípadom, kedy sa odpoveď S líši od odpovede B, je ak C pri svojich dotazoch na T nepoužije dešifrovací kľúč td odpovedajúci šifrovaciemu kľúču ek (hodnotu td vtedy simulátor S nemá ako zistiť, keďže pracujeme s permutáciou s padacími vrátkami). V tomto prípade vráti S ako svoju odpoveď hlášku `zlyhanie`. Aby sme dokázali, že S simuluje B nerozlíšiteľne až na zanedbateľné množstvo prípadov, musíme odhadnúť pravdepodobnosť javu, že C dokáže posledne uvedenú situáciu privodiť. Tomuto odhadu sa venujeme do konca dôkazu.

C môže ako odpoveď obdržať hlášku `zlyhanie` nasledujúcim spôsobom. C položí dotaz (k, ek, Π) simulátoru S^T, na základe čoho ako odpoveď obdrží $4k$ šifrových textov $[c_1, \ldots, c_{4k}]$. Tieto šifrové texty sú vypočítané podľa predpisu

$$c_i = f_{k,ek}(\pi(m_i||r_i))$$

s náhodne a uniformne volenými hodnotami m_i, r_i pre $i = 1, \ldots, 4k$. Následne C nejakým spôsobom správne určí všetky hodnoty m_i bez toho, aby mal k dispozícii inverziu permutácie $f_{k,ek}$. V opačnom prípade by totiž musel položiť dotaz na T s odpovedajúcim šifrovým kľúčom td využívajúc F^{-1}, čo urobiť nemôže, lebo simulátor S by takto získal hodnotu td vďaka svojmu prístupu ku všetkým takýmto dotazom podľa predpokladu Lemmy. C teda musí príslušné hodnoty m_i uhádnuť. Aby sme analyzovali pravdepodobnosť, že sa tak stane a preto S vráti ako svoj výstup hlášku `zlyhanie`, uvažujme nasledujúce tri množiny:

- Nech
$$\mathcal{R} = \{\pi(m||r) : m||r \in \{0,1\}^{\mu(k)+\rho(k)}\}$$
značí množinu všetkých vstupov, z ktorých je možné vypočítaním príslušnej funkčnej hodnoty $f_{k,ek}$ obdržať platný šifrový text. Keďže π je injektívne zobrazenie podľa definície schémy náhodného dopĺňania správy, platí
$$|\mathcal{R}| = 2^{\mu(k)+\rho(k)}.$$
- Nech
$$\mathcal{Y} = \{\pi(m_i||r) : i = 1, \ldots, 4k\}$$
značí množinu tých vstupov, z ktorých je možné vypočítaním príslušnej funkčnej hodnoty $f_{k,ek}$ obdržať šifrové texty c_i. Keďže π je injektívne zobrazenie, platí
$$|Y| = 4k.$$
- Nech
$$\mathcal{X} \subset \mathcal{R}$$
značí množinu všetkých dotazov, ktoré C položí T využívajúc $f_{k,ek}$ počas celého priebehu svojho výpočtu (teda pred aj po obdržaní vektoru šifrových textov $[c_1, \ldots, c_{4k}]$).

Označme ako
$$miss = |\mathcal{Y} \backslash \mathcal{X}|$$
počet možných vzorov šifrových textov c_i, na ktoré sa C behom svojho výpočtu nedotázalo. Vzory šifrových textov c_i sú v $\mathcal{R}$ distribuované uniformne a $f_{k,ek}$ je náhodná permutácia, takže $miss$ je náhodnou veličinou, ktorú môžeme modelovať nasledovne. Z množiny $\mathcal{R}$ veľkosti $2^{\mu(k)+\rho(k)}$ vyberieme nejakú náhodnú podmnožinu $\mathcal{X}$ veľkosti $q(k)$ a nejakú náhodnú podmnožinu $\mathcal{Y}$ veľkosti $4k$. Veličina $miss$ bude označovať počet prvkov množiny $\mathcal{Y}$, ktoré sa nevyskutyjú v množine $\mathcal{X}$. Stredná hodnota veličiny $miss$ vďaka náhodnosti a uniformnosti volených hodnôt splňuje
$$\mathrm{E}(miss) = 4k\left(1 - \frac{q(k)}{2^{\mu(k)+\rho(k)}}\right).$$
Keďže $q(k) \leq 2^{\mu(k)+\rho(k)}/2$ podľa predpokladu, platí
$$\mathrm{E}(miss) \geq \frac{4k}{2} = 2k.$$
Použitím Hoeffdingovej nerovnosti sa podľa [**12**] následne dopracujeme k poznatku, že pravdepodobnosť javu, že $miss$ nie je vzdialená od $\mathrm{E}(miss)$ o viac než k, je najmenej $1 - e^{-k/2}$. Tým pádom nadobúda $miss$ hodnotu aspoň k skoro iste, čo znamená, že C musí skoro iste uhádnuť aspoň k správ m_i, aby simulátor S vrátil ako svoj výstup hlášku `zlyhanie`. Pravdepodobnosť, že sa C podarí

uhádnuť aspoň k správ m_i, kde každá správa je volená náhodne a uniformne z $\{0,1\}^{\mu(k)}$, je najviac

$$\left(\frac{1}{2^{\mu(k)}}\right)^k = \frac{1}{2^{k\mu(k)}} \leq 2^{-k},$$

čo je funkcia zanedbateľná v k. Ukázali sme teda, že simulátor S simuluje výstupy orákula B presne až na zanedbateľné množstvo prípadov.

□

Práve dokázaná Lemma nám hovorí, že za určitých predpokladov je možné orákulum B efektívne simulovať. Tento fakt využijeme k dôkazu zovšeobecnenej verzie Lemmy 4.3.

LEMMA 4.6. TDP^T *je ideálnou permutáciou s padacími vrátkami s pravdepodobnosťou 1 cez všetky voľby orákul* T *a* B.

DÔKAZ. Uvažujme ľubovoľnú δ-ťažkú hru G a ľubovoľného pravdepodobnostného polynomiálneho útočníka A. Nech polynóm $q(k)$ označuje celkový počet dotazov realizovaných hrou G a útočníkom A dohromady. Keďže predpokladáme dostatočne veľký bezpečnostný parameter k, platí

$$(\forall^* k \in \mathbb{N}) : \left(q(k) \leq \frac{2^{\mu(k)+\rho(k)}}{2}\right).$$

Podľa Lemmy 4.5 teda môžeme orákulum B efektívne simulovať pomocou simulátora S, to znamená

$$\mathrm{Adv}^G_{\mathrm{TDP}^T}(A^{T,B}, k) \approx \mathrm{Adv}^G_{\mathrm{TDP}^T}(A^{T,S^T}, k)$$

Simulátor S je pravdepodobnostný polynomiálny algoritmus a preto môžeme jeho výpočet realizovať ako podprogram útočníka A. Nech teda $\hat{A}$ označuje útočníka A, ktorý si simuluje dotazy na S^T v rámci svojho vlastného výpočtu. Tým pádom je

$$\mathrm{Adv}^G_{\mathrm{TDP}^T}(A^{T,S^T}, k) = \mathrm{Adv}^G_{\mathrm{TDP}^T}(\hat{A}^T, k).$$

Podľa Lemmy 4.3 je TDP^T ideálnou permutáciou s padacími vrátkami cez všetky voľby T a preto je pravdepodobnosť výhry útočníka $\hat{A}$ v ľubovoľnej δ-ťažkej hre obmedzená výrazom

$$\mathrm{Adv}^G_{\mathrm{TDP}^T}(\hat{A}^T, k) \leq \delta + negl(k),$$

kde výrazom $negl(k)$ označujeme nejakú funkciu zanedbateľnú v k. Z vyššie odvodených vzťahov dostávame:

$$\mathrm{Adv}^G_{\mathrm{TDP}^T}(A^{T,B}, k) \leq \delta + negl'(k),$$

kde $negl'(k) \approx negl(k)$. To znamená, že ľubovoľná hra G je δ-ťažkou aj cez všetky voľby oboch orákul T a B.

□

Práve dokázanou Lemmou sme splnili posledný bod 2). Tým je dôkaz Tvrdenia 4.1 hotový.

□

4.2. Slabá neohybnosť plne inštancializovanej varianty OAEP

V tomto oddieli budeme čerpať z práce [**4**] z roku 2006, v ktorej dvojica autorov Boldyreva a Fischlin priniesla niekoľko pozitívnych výsledok týkajúci sa bezpečnosti OAEP v štandardnom modele.

4.2.1. Varianty OAEP a inštancializácia

Poznámka. Rodina funkcií $F = \bigcup_k F(1^k)$ je tvorená množinami funkcií $F(1^k) = \{f : \{0,1\}^{m(k)} \to \{0,1\}^{n(k)}\}$. Značením $f \leftarrow_{rnd} F(1^k)$ budeme rozumieť náhodnú voľbu funkcie f z množiny $F(1^k)$. Ak nebude uvedené inak, v tomto oddieli implicitne predpokladáme, že funkcia f je jednosmernou permutáciou s inverzom f^{-1}.

V ďalšom texte budeme pracovať s kryptosystémom OAEP, pričom budeme opäť vychádzať z jeho popisu uvedeného v oddieli 2.2.1. Podľa predchádzajúcej poznámky však mierne upravíme značenie. Kryptosystémom OAEP budeme rozumieť trojicu pravdepodobnostných polynomiálnych algoritmov $OAEP^{G,H}[F] = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, kde $\mathcal{K}$ označuje algoritmus pre generáciu kľúča produkujúci dvojicu $(f, f^{-1}) \leftarrow_{rnd} F(1^k)$ (verejný a privátny kľúč), $\mathcal{E}$ označuje algoritmus pre šifrovanie správ a nakoniec $\mathcal{D}$ označuje algoritmus pre ich dešifrovanie.

Ako sme si ukázali v oddieli 2.3.3., kryptosystém $OAEP^{G,H}[F]$ je bezpečný voči adaptívnemu útoku s voľbou šifrového textu za predpokladu, že príslušná šifrovacia transformácia je permutáciou, ktorá je na častiach definičného oboru jednosmerná. Vedľajším dôsledkom tohoto faktu je nasledujúce pozorovanie. Uvažujme variantu OAEP, kde namiesto rodiny jednosmerných permutácií s padacími vrátkami F použijeme rodinu $F_{t-clear}$, v ktorej je každá permutácia $f_{t-clear} : \{0,1\}^k \to \{0,1\}^k$ daná predpisom

$$f_{t-clear}(s||t) = f(s)||t,$$

pričom $f : \{0,1\}^{k-k_0} \to \{0,1\}^{k-k_0}$ je jednosmerná permutácia s padacími vrátkami z rodiny F. Náhodnú dvojicu vzájomne inverzných permutácií $(f_{t-clear}, f^{-1}_{t-clear}) \leftarrow_{rnd} F_{t-clear}(1^k)$ teda získame z náhodnej voľby $(f, f^{-1}) \leftarrow_{rnd} F(1^k)$ pomocou vyššie uvedeného predpisu pre $f_{t-clear}$. Takto definovaná rodina permutácií $F_{t-clear}$ obsahuje výhradne permutácie s padacími vrátkami, ktoré sú jednosmerné na častiach definičného oboru (v zmysle definície z oddielu 2.3.3.), a tak je kryptosystém $OAEP^{G,H}[F_{t-clear}]$ bezpečný voči adaptívnemu útoku s voľbou šifrového textu podľa Vety 2.3 – samozrejme iba v modele náhodného orákula, v ktorom je táto veta dokázaná.

Uvažujme ďalšiu variantu OAEP, kde oproti predchádzajúcej variante bude príslušná funkcia vracať spolu s hodnotou t v otvorenej podobe navyše aj k_1 najnižších bitov hodnoty s, ktoré budeme označovať ako $lsb_{k_1}(s)$. Na zvyšných najvyšších $k - k_0 - k_1$ bitov hodnoty s aplikujeme jednosmernú permutáciu s padacími vrátkami f. Takže náhodná permutácia $f_{lsb||t-clear} : \{0,1\}^k \to \{0,1\}^k$ z rodiny permutácií $F_{lsb||t-clear}(1^k)$ je určená nejakou náhodnou jednosmernou permutáciou s padacími vrátkami $f : \{0,1\}^{k-k_0-k_1} \to \{0,1\}^{k-k_0-k_1}$ z rodiny F a nasledujúcim

predpisom:

$$f_{lsb||t-clear}(s||t) = f(s[k-k_0-1,\ldots,k_1])||lsb_{k_1}(s)||t.$$

Z platnosti vzťahu $s = G(r)\oplus(M||0^{k_1})$ dostávame $lsb_{k_1}(s) = lsb_{k_1}(G(r))$. Označme $s[k-k_0-1,\ldots,k_1] = s_{k-k_0-k_1}$ a $lsb_{k_1}(G(r)) = \gamma$. Dosadením získame vzťahy $s = s_{k-k_0-k_1}||\gamma$ a tiež

$$f_{lsb||t-clear}(s||t) = f(s_{k-k_0-k_1})||\gamma||t.$$

Analogickou úvahou ako v predchádzajúcom odstavci dostávame, že rodina permutácií $F_{lsb||t-clear}$ obsahuje tiež výhradne permutácie s padacími vrátkami, ktoré sú jednosmerné na častiach definičného oboru, a preto je aj kryptosystém $OAEP^{G,H}[F_{lsb||t-clear}]$ bezpečný voči adaptívnemu útoku s voľbou šifrového textu (v modele náhodného orákula).

Definícia (indukované rodiny permutácií). Rodiny permutácií $F_{t-clear}$ a $F_{lsb||t-clear}$ z predchádzajúcich odstavcov nazývame rodinami *indukovanými* rodinou F.

Definícia (inštancializácia náhodných orákul pomocou pseudonáhodných generátorov). Uvažujme dvojicu pravdepodobnostných polynomiálnych algoritmov $\mathcal{G} = (\mathrm{KGenG}, \mathrm{G})$, kde algoritmus KGenG na vstupe 1^k vracia ako výstup náhodný kľúč $K = \mathrm{KGenG}(1^k)$ a algoritmus G na vstupe (K, r), kde $r \in \{0,1\}^{k_0}$ je náhodná hodnota, vracia ako výstup hodnotu $\mathrm{G}_K(r) = \mathrm{G}(K, r)$ o dĺžke $k - k_0$ bitov. Zápisom $OAEP^{\mathcal{G},H}[F]$ rozumieme taký kryptosystém OAEP, v ktorom je každá hodnota $G(r)$ nahradená hodnotou $\mathrm{G}_K(r)$ a dvojicu odpovedajúcich kľúčov (pk, sk) (po poradí verejný a privátny kľúč) tvoria prvky

$$pk = (f, K),\ \ sk = (f^{-1}, K).$$

Vtedy hovoríme, že kryptosystém $OAEP^{\mathcal{G},H}[F]$ je *čiastočnou G-inštancializáciou OAEP realizovanou pseudonáhodným generátorom $\mathcal{G}$*. Podobne definujeme aj kryptosystém $OAEP^{G,\mathcal{H}}[F]$ ako *čiastočnú H-inštancializáciu OAEP realizovanú pseudonáhodným generátorom $\mathcal{H}$*. Nakoniec, ak popísaným spôsobom inštancializujeme obe orákulá G, H súčasne, nazývame kryptosystém $OAEP^{\mathcal{G},\mathcal{H}}[F]$ *plnou inštancializáciou OAEP realizovanou pseudonáhodnými generátormi $\mathcal{G}$ a $\mathcal{H}$.*

4.2.2. Slabá neohybnosť

Tento oddiel je venovaný zavedeniu pojmu neohybnosť (anglicky non-malleability) a jeho slabej verzie. Neohybnosť je jednou zo žiadúcich bezpečnostných vlastností kryptosystémov s verejným kľúčom. Základná verzia neohybnosti sa označuje ako NM-CPA (z anglického non-malleable chosen plaintext attack) a hovorí, že znalosť šifrového textu C^* odpovedajúceho cieľovej správe M^* nesmie útočníkovi významným spôsobom pomôcť určiť šifrový text C odpovedajúci súvisiacej správe M. Rozdelenie cieľovej správy M^* je určené nejakým rozdelením $\mathcal{M}$. Súvislosť správ M^* a M je daná polynomiálnou binárnou reláciou R. Rozdelenie $\mathcal{M}$ aj popis relácie R pritom určuje útočník $\mathcal{A}$.

Definícia (neohybnosť). Nech $\mathrm{AS} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ je asymetrický kryptosystém. Uvažujme nasledujúce scenáre a náhodné veličiny popisujúce ich výsledok:

- náhodná veličina $\mathrm{Exp}^{nm-cpa-1}_{\mathrm{AS},\mathcal{A}}(k) \in \{0,1\}$:

$$\mathcal{K}(1^k) = (pk, sk)$$

$$\mathcal{A}(pk) = \mathcal{M}$$

$$M^* \leftarrow_{rnd} \mathcal{M}$$

$$\mathcal{E}_{pk}(M^*) = C^*$$

$$\mathcal{A}(C^*) = (R, C)$$

$$\mathcal{D}_{sk}(C) = M$$

$$\left(\mathrm{Exp}_{\mathrm{AS},\mathcal{A}}^{nm-cpa-1}(k) = 1\right) \Leftrightarrow ((C^* \neq C) \ \&\ (M^*, M) \in R)$$

- náhodná veličina $\mathrm{Exp}_{\mathrm{AS},\mathcal{A}}^{nm-cpa-0}(k) \in \{0,1\}$:

$$\mathcal{K}(1^k) = (pk, sk)$$

$$\mathcal{A}(pk) = \mathcal{M}$$

$$M^* \leftarrow_{rnd} \mathcal{M};\ M' \leftarrow_{rnd} \mathcal{M}$$

$$\mathcal{E}_{pk}(M') = C'$$

$$\mathcal{A}(C') = (R, C)$$

$$\mathcal{D}_{sk}(C) = M$$

$$\left(\mathrm{Exp}_{\mathrm{AS},\mathcal{A}}^{nm-cpa-0}(k) = 1\right) \Leftrightarrow ((C' \neq C) \ \&\ (M^*, M) \in R)$$

Asymetrický kryptosystém AS nazývame *bezpečným v zmysle NM-CPA* alebo tiež *neohybným*, ak pre ľubovoľný pravdepodobnostný polynomiálny algoritmus $\mathcal{A}$ sú náhodné veličiny $\mathrm{Exp}_{\mathrm{AS},\mathcal{A}}^{nm-cpa-1}(k)$ a $\mathrm{Exp}_{\mathrm{AS},\mathcal{A}}^{nm-cpa-0}(k)$ výpočtovo nerozlíšiteľné (ich rozlíšenie ako dvoch štatistických súborov je výpočtovo nedosažiteľnou úlohou).

Slabú verziu neohybnosti budeme označovať ako RNM-CPA. Od neohybnosti kryptosystému, ktorú sme si práve definovali, sa slabá neohybnosť líši tým, že útočník nemá možnosť voliť rozdelenie $\mathcal{M}$ cieľovej správy M^* ani reláciu R. Rozdelenie je tu pevne dané ako uniformné na všetkých bitových reťazcoch stanovenej dĺžky a pevne daná je aj súvislosť cieľovej správy M^* so správou M prostredníctvom fixovanej relácie R, ktorá je stanovená dopredu.

Definícia (slabá neohybnosť). Nech $\mathrm{AS} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ je asymetrický kryptosystém. Ďalej nech $\mathcal{M}(1^k)$ označuje uniformné rozdelenie na všetkých bitových reťazcoch dĺžky $\ell(k)$ pre nejaký polynóm ℓ. Uvažujme nasledujúce scenáre a náhodné veličiny popisujúce ich výsledok:

- náhodná veličina $\mathrm{Exp}_{\mathrm{AS},\mathcal{A},\mathcal{M},R}^{rnm-cpa-1}(k) \in \{0,1\}$:

$$\mathcal{K}(1^k) = (pk, sk)$$

$$M^* \leftarrow_{rnd} \mathcal{M}(1^k)$$

$$\mathcal{E}_{pk}(M^*) = C^*$$

$$\mathcal{A}(C^*, pk, R) = C$$

$$\mathcal{D}_{sk}(C) = M$$

$$\left(\mathrm{Exp}_{\mathrm{AS},\mathcal{A},\mathcal{M},R}^{rnm-cpa-1}(k) = 1\right) \Leftrightarrow ((C^* \neq C) \ \&\ (M^*, M) \in R)$$

- náhodná veličina $\mathrm{Exp}^{rnm-cpa-0}_{\mathrm{AS},\mathcal{A},\mathcal{M},R}(k) \in \{0,1\}$:

$$\mathcal{K}(1^k) = (pk, sk)$$
$$M^* \leftarrow_{rnd} \mathcal{M}(1^k);\ M' \leftarrow_{rnd} \mathcal{M}(1^k)$$
$$\mathcal{E}_{pk}(M') = C'$$
$$\mathcal{A}(C', pk, R) = C$$
$$\mathcal{D}_{sk}(C) = M$$
$$\left(\mathrm{Exp}^{rnm-cpa-0}_{\mathrm{AS},\mathcal{A},\mathcal{M},R}(k) = 1\right) \Leftrightarrow ((C' \neq C)\ \&\ (M^*, M) \in R)$$

Asymetrický kryptosystém AS nazývame *bezpečným v zmysle RNM-CPA* alebo tiež *slabo neohybným*, ak pre ľubovoľný pravdepodobnostný polynomiálny algoritmus $\mathcal{A}$ a ľubovoľnú polynomiálnu binárnou reláciu R sú náhodné veličiny $\mathrm{Exp}^{nm-cpa-1}_{\mathrm{AS},\mathcal{A}}(k)$ a $\mathrm{Exp}^{nm-cpa-0}_{\mathrm{AS},\mathcal{A}}(k)$ výpočtovo nerozlíšiteľné (ich rozlíšenie ako dvoch štatistických súborov je výpočtovo nedosažiteľnou úlohou).

Slabá neohybnosť kvôli vyššie uvedeným rozdielom oproti plnej verzii garantuje skutočne slabšiu úroveň bezpečnosti kryptosystému, napriek tomu však má svoj význam. Podľa [**4**] postačuje pre dokázanie bezpečnosti kryptosystému v zmysle OW-CPA, kde cieľom útočníka je rekonštrukcia celej správy z ľubovoľne zadaného cieľového šifrového textu (útočník má pritom k dispozícii šifrovacie orákulum). Slabá neohybnosť rovnako postačuje pre vylúčenie Bleichenbacherovho útoku na PKCS #1 v1.5. V nasledujúcich častiach tohto oddielu si ukážeme, že bezpečnosť kryptosystému v zmysle RNM-CPA je za určitých podmienok dostačujúca pre odvodenie bezpečnosti v zmysle NM-CPA, pričom ako je známe (viď napríklad práca [**1**]), tento výsledok implikuje aj bezpečnosť v zmysle IND-CPA. Konkrétne, zvolí sa náhodná hodnota r, ktorá sa zašifruje asymetrickým kryptosystémom bezpečným v zmysle RNM-CPA. Následne sa využije náhodné orákulum G k odvodeniu kľúča $K = G(r)$ pre symetrický kryptosystém. Príkladom využitia je rozšírený komunikačný protokol SSL, ktorý sa do istej miery dá vnímať ako špeciálny prípad tejto metódy. V protokole SSL klient posiela serveru zašifrovanú náhodnú hodnotu, z ktorej si následne obe strany odvodia symetrický kľúč pomocou komplikovanej hashovacej operácie. Uvedené využitie bezpečnosti kryptosystému v zmysle RNM-CPA k získaniu bezpečnosti v zmysle NM-CPA (a teda aj IND-CPA) spoločne s dôkazom bezpečnosti plne inštancializovaného kryptosystému $\mathrm{OAEP}^{\mathcal{G},\mathcal{H}}[F]_{lsb||t-clear}$ v zmysle RNM-CPA, ktorý je hlavným cieľom tohto oddielu, teda za predpokladu ideálnej generácie kľúča poskytuje zaujímavý výsledok. Tento výsledok je v súčasnosti najsilnejším známym pozitívnym výsledkom týkajúcim sa bezpečnosti OAEP v štandardnom modele.

4.2.3. Pseudonáhodné generátory

Minimálnou požiadavkou, ktorej splnenie pri inštancializácii náhodného orákula pseudonáhodným generátorom typicky očakávame, je aby algoritmus realizujúci inštancializáciu obsahoval popis pseudonáhodného generátora. Pseudonáhodný generátor pozostáva z algoritmu KGen pre generáciu verejného kľúča K a z algoritmu G transformujúceho náhodnú hodnotu (semienko) r spoločne s kľúčom K na pseudonáhodný výstup. Tento výstup obvykle ostáva z vonkajšieho pohľadu náhodným aj po sprístupnení postrannej informácie $\mathrm{hint}(r)$ o semienku r, kde hint značí nejaký pravdepodobnostný polynomiálny algoritmus, ktorého výstup je ťažko invertovateľný.

Nižšie uvedená definícia pseudonáhodného generátora taktiež zahrňuje aj možnosť, že výstupom algoritmu pre generáciu kľúča je okrem K ešte aj nejaká tajná informácia K^{-1} (padacie vrátka). Použitím padacích vrátok K^{-1} je možné efektívne invertovať výstup pseudonáhodného generátora a získať tak semienko r. Ak padacie vrátka nemajú žiadne využitie (nemá zmysel ich generovať), položíme buď $K^{-1} = \perp$, prípadne hodnotu K^{-1} z výstupu algoritmu pre generáciu kľúča úplne vynechávame.

Definícia (pseudonáhodný generátor). Nech KGen je pravdepodobnostný polynomiálny algoritmus pre generáciu kľúča, ktorého vstupom je hodnota 1^k pre bezpečnostný parameter $k \in \mathbb{N}$ a výstupom je kľúč K. Ďalej nech G je deterministický polynomiálny algoritmus, ktorého vstupom je kľúč K a náhodná hodnota $r \in \{0,1\}^k$ a výstupom je bitový reťazec dĺžky $\ell(k)$. Potom $\mathcal{G} = (\mathrm{KGen}, \mathrm{G})$ sa nazýva *pseudonáhodný generátor (vzhľadom k hint)*, ak sú nasledujúce náhodné vektory výpočtovo nerozlíšiteľné:

- $\mathrm{KGen}(1^k) = K$, $r \leftarrow_{rnd} \{0,1\}^k$, $\mathrm{hint}(r) = h$, náhodný vektor $(K, \mathrm{G}(K,r), h)$,
- $\mathrm{KGen}(1^k) = K$, $r \leftarrow_{rnd} \{0,1\}^k$, $\mathrm{hint}(r) = h$, $u \leftarrow_{rnd} \{0,1\}^{\ell(k)}$, náhodný vektor (K, u, h).

Ak navyše existuje polynomiálny algoritmus TdG, ktorý pre každé $k \in \mathbb{N}$, každú dvojicu $\mathrm{KGen}(1^k) = (K, K^{-1})$ a každé $r \in \{0,1\}^k$ spĺňuje rovnosť

$$\mathrm{G}(K, \mathrm{TdG}(K^{-1}, \mathrm{G}(K,r))) = \mathrm{G}(K,r),$$

potom trojicu $(\mathrm{KGen}, \mathrm{G}, \mathrm{TdG})$ nazývame *pseudonáhodným generátorom s padacími vrátkami*.

Pre účely tohto oddielu budeme potrebovať pseudonáhodné generátory spĺňajúce niektoré ďalšie vlastnosti ako napríklad bezkolíznosť alebo neohybnosť. Bezkolíznosť v prípade pseudonáhodných generátorov znamená, že k danému semienku r je ťažké nájsť iné semienko r', ktoré by spĺňovalo rovnosť $\mathrm{G}(K,r) = \mathrm{G}(K,r')$ na dopredu určenej množine bitov. Neohybnosť zase zaisťuje, aby výstup pseudonáhodného generátora pre dané semienko r nebol významným spôsobom užitočný k vytvoreniu iného výstupu pre semienko r' nejakým spôsobom súvisiace so semienkom r. Presné definície uvedených pojmov zavedieme ďalej v tomto oddieli.

4.2.4. Inštancializácia náhodného orákula G

V tomto a nasledujúcom oddieli si ukážeme, že čiastočne inštancializované OAEP, kedy inštancializujeme práve jedno z náhodných orákul G a H, si zachováva bezpečnosť voči adaptívnemu útoku s voľbou šifrového textu (v zmysle CCA2). Keďže druhé náhodné orákulum ostáva v takom prípade neinštancializované, hovoríme stále o preukázateľnej bezpečnosti v modele náhodného orákula.

Najskôr si ukážeme, ako je možné skonštruovať pseudonáhodný generátor so špeciálnou formou bezkolíznosti. Táto vlastnosť znamená, že nájsť hdnotu r' k náhodnej hodnote r tak, aby sa reťazce $\mathrm{G}(K,r)$ a $\mathrm{G}(K,r')$ zhodovali na k najnižších bitoch $\mathrm{lsb}_k(\mathrm{G}(K,r))$ a $\mathrm{lsb}_k(\mathrm{G}(K,r'))$, je výpočtovo nedosažiteľné. Táto vlastnosť pripomína bezkolíznosť hashovacích funkcií, vďaka čomu má aj podobné pomenovanie.

Definícia (bezkolízny pseudonáhodný generátor). Pseudonáhodný generátor $\mathcal{G} = (\mathrm{KGen}, \mathrm{G})$ nazývame *bezkolíznym (vzhľadom k najnižším k bitom)*, ak pre ľubovoľný pravdepodobnostný polynomiálny algoritmus C platí nasledujúce. Nech $\mathrm{KGen}(1^k) = K$, $r \leftarrow_{rnd} \{0,1\}^k$, $C(K, r) = r'$. Potom pravdepodobnosť, že

$$(r \neq r') \ \& \ (\mathrm{lsb}_k(\mathrm{G}(K, r)) = \mathrm{lsb}_k(\mathrm{G}(K, r')))$$

je zanedbateľnou funkciou v k.

Bezkolízne pseudonáhodné generátory môžu byť podľa [**4**] vytvorené napríklad pomocou jednosmerných permutácií využitím konštrukcie Yao-Blum-Micali (skrátene YBM). V tomto prípade je daná nejaká rodina jednosmerných permutácií G. Algoritmus pre generáciu kľúča $\mathrm{KGen}_{YBM}(1^k)$ zvolí ako kľúč náhodnú permutáciu $g : \{0,1\}^k \to \{0,1\}^k$ z množiny $G(1^k)$. Algoritmus G_{YBM} vráti ako svoj výstup $\mathrm{G}_{YBM}(g, r) = (\mathrm{hb}(r), \mathrm{hb}(g(r)), \ldots, \mathrm{hb}(g^{n-1}(r)), g^n(r))$, ktorého všetky bity až na k najnižších, tvoria hardcore bity hb permutácie g. Najnižších k bitov výstupu $\mathrm{G}_{YBM}(g, r)$ tvorí hodnota $g^n(r)$. Keďže g je permutácia, rozdielne vstupy $r \neq r'$ zobrazuje na rozdielne výstupy $g^n(r) \neq g^n(r')$. Preto je uvedená konštrukcia $\mathcal{G}_{YBM} = (\mathrm{KGen}_{YBM}, \mathrm{G}_{YBM})$ podľa predchádzajúcej definície bezkolíznym pseudonáhodným generátorom.

Pomocou bezkolíznych pseudonáhodných generátorov je možné inštancializovať náhodné orákulum G v kryptosystéme $\mathrm{OAEP}^{G,H}[F_{t-clear}]$ pre rodinu $F_{t-clear}$ indukovanú rodinou permutácií s padacími vrátkami F (viď oddiel 4.2.1.). Ukážeme si, ako.

Veta 4.7. *Nech $\mathcal{G} = (KGen, G)$ je bezkolízny pseudonáhodný generátor vzhľadom k najnižším k_1 bitom. Ďalej nech F je rodina permutácií s padacími vrátkami a nech $F_{t-clear}$ je rodina permutácií s padacími vrátkami jednosmerných na častiach definičného oboru indukovaná F z oddielu 4.2.1. Potom $OAEP^{\mathcal{G},H}[F_{t-clear}]$, čiže čiastočná G-inštancializácia varianty OAEP realizovaná $\mathcal{G}$, je bezpečná voči adaptívnemu útoku s voľbou šifrového textu v modele náhodného orákula.*

Náznak dôkazu. Dôkaz využíva Shoupovu metodiku hier (viď oddiel 2.3.2.). Hlavnou myšlienkou je postupné upravovanie spôsobu, akým je generovaný cieľový šifrový text. Ako je ukázané v práci [**4**], pravdepodobnosť útočníkovho úspechu sa vo všetkých nasledujúcich hrách líši vždy iba o zanedbateľnú funkciu:

- v počiatočnej hre Hra^0 je cieľový šifrový text $f(s^*)||t^*$ pre správu M^* určený podľa definície krytposystému $OAEP^{\mathcal{G},H}[F_{t-clear}]$, teda

  $$s^* = \mathrm{G}(K, r^*) \oplus (M^*||0^{k_1})$$

  pre bezkolízny pseudonáhodný generátor G a

  $$t^* = H(s^*) \oplus r^*$$

  pre náhodné orákulum H.
- v nasledujúcej hre Hra^1 je hodnota

  $$s^* = \mathrm{G}(K, r^*) \oplus (M^*||0^{k_1})$$

  počítaná ako v predchádzajúcej hre, ale hodnotu t^* počítame takto:

  $$t^* = \omega \oplus r^*,$$

kde ω je náhodne a uniformne volená hodnota, ktorá je nezávislá na $H(s^*)$. Keďže H je náhodné orákulum, táto zmena je z pohľadu útočníka nerozlíšiteľná (výsledné rozdelenie cieľového šifrového textu sa nemení). Jediným patologickým prípadom je situácia, kedy sa útočník v priebehu svojho výpočtu dotáže orákula H na hodnotu s^*, čo vedie k inkonzistencii. Pravdepodobnosť tohto javu je však zanedbateľná, konkrétne $1/2^{k-k_0}$ (bitová dĺžka s^* je práve $k - k_0$).

- v ďalšej hre Hra^2 upravíme výpočet hodnoty r^* nasledovne:

$$t^* = \omega,$$

čím odstránime závislosť hodnoty t^* na hodnote r^*. Rozdelenie t^* ostáva nezmenené, pretože hodnota ω je podľa predchádzajúcej odrážky volená náhodne a uniformne (rovnako ako bola aj hodnota r^*).

- v poslednej hre Hra^3 využijeme pseudonáhodnosť generátora G k zámene hodnoty $s^* = \mathrm{G}(K, r^*) \oplus (M^*||0^{k_1})$ za hodnotu

$$s^* = u \oplus (M^*||0^{k_1}),$$

kde u je hodnota volená náhodne a uniformne.

Rozborom poslednej hry Hra^3 dospejeme k záveru, že dotazy na dešifrovacie orákulum útočníkovi pri jeho výpočte nepomôžu. Využijeme k tomu práve bezkolíznosť pseudonáhodného generátora $\mathcal{G}$. Konkrétne, bezkolíznosť zabraňuje útočníkovi v transformácii cieľového šifrového textu odpovedajúceho hodnotám r^* a s^* na iný platný šifrový text s rovnakým s^*, ale odlišným r. Dôvodom je, že najnižších k_1 bitov $s^* = \mathrm{G}(K, r^*) \oplus (M^*||0^{k_1}) = \mathrm{G}(K, r) \oplus (M||0^{k_1})$ sa podľa definície bezkolízneho pseudonáhodného generátora nemôže zhodovať s pravdepodobnosťou, ktorá nie je zanedbateľná. Tým pádom bude takto odvodený šifrový text s vysokou pravdepodobnosťou neplatný. Preto každý platný šifrový text, na ktorý sa útočník dotáže dešifrovacieho orákula, musí obsahovať hodnotu s rozdielnu od s^* a útočník sa pred jeho vytvorením musí dotázať náhodného orákula na hodnotu $H(s)$ (inak by vytvorenie platného šifrového textu mohlo nastať iba so zanedbateľnou pravdepodobnosťou – dešifrovacie orákulum totiž pri korektnom dešifrovaní volá náhodné orákulum H za účelom získania hodnoty $H(s)$). Lenže v tomto prípade útočník ešte pred vytvorením šifrového textu musí poznať hodnotu $r = t \oplus H(s)$, lebo hodnotu t pozná vďaka konštrukcii $F_{t-clear}$ (v šifrovom texte sa táto hodnota objavuje v otvorenej podobe). Útočník teda ešte pred vytvorením šifrového textu musí poznať aj odpovedajúcu hodnotu $M||z = s \oplus \mathrm{G}(K, r)$, pretože hodnota K je súčasťou verejného kľúča. Tým sme ukázali, že dešifrovacie orákulum útočníkovi pri útoku nepomôže, lebo výslednú správu M si útočník dokáže spočítať aj sám.

□

4.2.5. Inštancializácia náhodného orákula H

Aby sa nám podarilo dokázať bezpečnosť varianty OAEP s inštancializovaným náhodným orákulom H, zavedieme si pojem neohybného pseudonáhodného generátora. Pre takýto generátor je výpočtovo nedosažiteľné k danej hodnote $y^* = \mathrm{H}_K(s^*)$ pre náhodné s^* nájsť inú (rozdielnu) hodnotu $y = \mathrm{H}_K(s)$ pre s nejakým spôsobom súvisiacim s s^*, pričom odpovedajúca polynomiálna relácia R je

definovaná ešte pred obdržaním hodnôt K a y^*. Presnejšie, neohybnosť pseudonáhodného generátora je formálne definovaná ako nerozlíšiteľnosť dvoch experimentov. Ľubovoľný pravdepodobnostný polynomiálny útočník $\mathcal{B}$ by nemal byť schopný rozlíšiť situáciu, kedy obdrží hodnoty $f(s^*)$ a $y^* = \mathrm{H}_K(s^*)$ od situácie, kedy namiesto nich obdrží hodnoty $f(s^*)$ a $y' = \mathrm{H}_K(s')$ pre s' volené nezávisle na s^*. Pravdepodobnosť, že $\mathcal{B}$ vráti ako svoj výstup hodnoty $f(s)$ a $y = \mathrm{H}_K(s)$ také, že $(s^*, s) \in R$ je splnená, by sa mala v oboch prípadoch líšiť nanajvýš o zanedbateľnú funkciu.

Definícia (neohybný pseudonáhodný generátor). Nech $\mathcal{H} = (\mathrm{KGenH}, \mathrm{H})$ je pseudonáhodný generátor vzhľadom k $\mathrm{hint}(x) = (f, f(x))$ pre $(f, f^{-1}) \leftarrow_{rnd} F(1^k)$, kde F je nejaká rodina permutácií s padacími vrátkami. Potom $\mathcal{H}$ sa nazýva *neohybný pseudonáhodný generátor vzhľadom k* hint, ak pre každý pravdepodobnostný polynomiálny algoritmus $\mathcal{B}$ a každú binárnu polynomiálnu reláciu R sú nasledujúce dve náhodné veličiny výpočtovo nerozlíšiteľné:

- náhodná veličina $\mathrm{Exp}^{nm-cpa-1}_{\mathcal{G},\mathcal{B},F,R}(k) \in \{0,1\}$:

$$\mathrm{KGenH}(1^k) = K$$

$$(f, f^{-1}) \leftarrow_{rnd} F(1^k)$$

$$s^* \leftarrow_{rnd} \{0,1\}^k$$

$$\mathrm{H}_K(s^*) = y^*$$

$$\mathcal{B}(K, f, f(s^*), y^*) = (z, y)$$

$$f^{-1}(z) = s$$

$$\left(\mathrm{Exp}^{nm-cpa-1}_{\mathcal{G},\mathcal{B},F,R}(k) = 1\right) \Leftrightarrow ((s^*, s) \in R \;\&\; (s^* \neq s) \;\&\; (\mathrm{H}_K(s) = y))$$

- náhodná veličina $\mathrm{Exp}^{nm-cpa-0}_{\mathcal{G},\mathcal{B},F,R}(k) \in \{0,1\}$:

$$\mathrm{KGenH}(1^k) = K$$

$$(f, f^{-1}) \leftarrow_{rnd} F(1^k)$$

$$s^* \leftarrow_{rnd} \{0,1\}^k;\; s' \leftarrow_{rnd} \{0,1\}^k$$

$$\mathrm{H}_K(s') = y'$$

$$\mathcal{B}(K, f, f(s^*), y') = (z, y)$$

$$f^{-1}(z) = s$$

$$\left(\mathrm{Exp}^{nm-cpa-0}_{\mathcal{G},\mathcal{B},F,R}(k) = 1\right) \Leftrightarrow ((s^*, s) \in R \;\&\; (s^* \neq s) \;\&\; (\mathrm{H}_K(s) = y))$$

Pomocou neohybného pseudonáhodného generátora môžeme dokázať bezpečnosť čiastočnej H-inštancializácie kryptosystému OAEP v zmysle NM-CPA za predpokladu, že relácia R a rozdelenie útočníkom volených správ použitých pri útoku sú dané ako $\mathcal{A}(1^k) = (\mathcal{M}, R)$ ešte pred zahájením samotného útoku a závisia tak iba na bezpečnostnom parametri k. Podľa [**4**] je tento výsledok stále postačujúci k odvodeniu bezpečnosti v zmysle IND-CPA (pre správy volené nezávisle na verejnom kľúči) a tiež zamedzuje napríklad Bleichenbacherovmu útoku.

VETA 4.8. *Nech F je rodina permutácií s padacími vrátkami a nech $F_{t-clear}$ je rodina permutácií s padacími vrátkami jednosmerných na častiach definičného oboru indukovaná F z oddielu 4.2.1. Ďalej nech $\mathcal{H} = (\mathrm{KGenH}, \mathrm{H})$ je pseudonáhodný generátor vzhľadom k hint$(x) = (f, f(x))$ pre $(f, f^{-1}) \leftarrow_{rnd} F(1^k))$, ktorý je neohybný vzhľadom k hint. Potom $OAEP^{G,\mathcal{H}}[F_{t-clear}]$, čiže čiastočná H-inštancializácia varianty OAEP realizovaná $\mathcal{H}$, je bezpečná v zmysle NM-CPA v modele náhodného orákula za predpokladu, že útočník $\mathcal{A}$ definuje $\mathcal{A}(1^k) = (\mathcal{M}, R)$ ešte pred zahájením samotného útoku.*

NÁZNAK DÔKAZU. Útočník obdrží cieľový šifrový text odpovedajúci nejakým hodnotám r^* a s^*. Tieto hodnoty jednoznačne určujú príslušnú správu M^*. Predpokladajme, že sa útočník pokúsi vyprodukovať platný šifrový text pre nejakú hodnotu $r \neq r^*$ bez toho, aby sa najskôr na hodnotu r dotázal náhodného orákula G. Potom s pravdepodobnosťou, ktorá sa od 1 líši iba o zanedbateľnú funkciu, nie je posledných k_1 bitov hodnoty $s \oplus G(r)$ nulových, čo znamená neplatnosť príslušného šifrového textu. V opačnom prípade, teda ak $r = r^*$, plynie z konštrukcie šifrovacej transformácie $OAEP^{G,\mathcal{H}}[F_{t-clear}]$, že $f(s) \neq f(s^*)$ a tým pádom aj $s \neq s^*$ (f je permutácia), pretože vyprodukovaný šifrový text sa musí líšiť od cieľového šifrového textu. Ostáva teda jediná možnosť k zlomeniu NM-CPA bezpečnosti daného kryptosystému – útočník by musel vedieť nejakým spôsobom definovať reláciu, ktorú by hodnoty $\mathrm{H}_K(s^*)$ a $\mathrm{H}_K(s)$ splňovali a ktorá by bola odvoditeľnú z relácie R určenej útočníkom pre šifrové texty. To je ale spor s predpokladom neohybnosti pseudonáhodného generátora $\mathcal{H}$. Pre úplnosť si rozoberme aj prípad, kedy $r \neq r^*$ a útočník sa na hodnotu r dotazuje náhodného orákula G. Vtedy je náhodná hodnota $G(r^*)$ nezávislá od náhodnej hodnoty $G(r^*)$ a preto sú navzájom nezavislé aj odpovedajúce správy $M^*||0^{k_1} = s^* \oplus G(r^*)$ a $M||0^{k_1} = s \oplus G(r)$, čo je pre potrebné k dosiahnutiu požadovanej bezpečnosti v zmysle NM-CPA. □

Inštancializácia náhodného orákula H je podľa [**4**] možná aj takým spôsobom, aby sme dosiahli bezpečnosť v zmysle CCA2 ako pri inštancializácii náhodného orákula G v predchádzajúcom oddieli. Toto ale vyžaduje veľmi silné predpoklady na použitý pseudonáhodný generátor. Konkrétne, útočníkovi by sme museli povoliť dotazy na inverz výstupu generátora (samozrejme so zachovaním ostatných vlastností tohto generátora). Taký pseudonáhodný generátor je ale tým pádom už porovnateľný s kryptosystémom bezpečným voči útoku s voľbou šifrového textu. Preto má práve dokázaná Veta význam skôr ako indikácia, že čiastočná H-inštancializácia $OAEP^{G,H}[F_{t-clear}]$ je principiálne možná.

4.2.6. Plná inštancializácia

V tomto oddieli si ukážeme plnú inštancializáciu kryptosystému $OAEP_{lsb||clear}$ bezpečnú v zmysle RNM-CPA v štandardnom modele. Pripomeňme, že podľa značenia z oddielu 4.2.1. vo variante $OAEP_{lsb||clear}$ píšeme $s_{k-k_0-k_1}||\gamma = G(r) \oplus (M||0^{k_1})$.

Aby sme dokázali, čo chceme, budeme potrebovať bezkolízny pseudonáhodný generátor s padacími vrátkami. Taký generátor je možné získať nasledujúcou úpravou konštrukcie Yao-Blum-Micali z oddielu 4.2.4. Jednosmerná permutácia g bude tentokrát permutáciou s padacími vrátkami. Algoritmus G_{YBM} vráti opäť ako svoj výstup $\mathrm{G}_{YBM}(g, r) = (\mathrm{hb}(r), \mathrm{hb}(g(r)), \ldots, \mathrm{hb}(g^{n-1}(r)), g^n(r))$. Ak bude

K^{-1}, ktoré je súčasťou výstupu KGen, obsahovať ako padacie vrátka inverznú permutáciu g^{-1}, potom algoritmus TdG dokáže jednoducho invertovať k_1 najnižších bitov výstupu G_{YBM}, ktoré su tvorené hodnotou $g^n(r)$. Tým získa hodnotu r.

O generátore G_{YBM} budeme predpokladať, že algoritmus TdG dokáže získať hodnotu r aj keď obdrží iba najnižších k_1 bitov výstupu G_{YBM} (bez prístupu k zvyšnej časti výstupu). Navyše budeme ešte predpokladať, že týchto k_1 najnižších bitov výstupu G_{YBM} má rozdelenie štatisticky blízke uniformnému rozdeleniu na $\{0,1\}^k$. Nakoniec, asi najzásadnejším predpokladom je, že správy odpovedajúce platným šifrovým textom, na ktoré sa útočník behom útoku dotazuje dešifrovacieho orákula, sú volené náhodne a uniformne (tj. rozdelenie $\mathcal{M}$ je uniformné) a relácia R je daná ešte pred začiatkom samotného útoku.

Veta 4.9. *Nech F je rodina permutácií s padacími vrátkami a nech $F_{lsb||t-clear}$ je rodina permutácií s padacími vrátkami jednosmerných na častiach definičného oboru indukovaná F z oddielu 4.2.1. Ďalej nech $\mathcal{G} = (KGenG, G)$ je bezkolízny pseudonáhodný generátor s padacími vrátkami vzhľadom k najnižším k_1 bitom. Nakoniec, nech $\mathcal{H} = (KGenH, H)$ je neohybný pseudonáhodný generátor vzhľadom k hint$(s_{k-k_0-k_1}||\gamma) = (f, f(s_{k-k_0-k_1}||\gamma)$ pre $(f, f^{-1}) \leftarrow_{rnd} F(1^k)$. Potom plná inštancializácia $OAEP^{\mathcal{G},\mathcal{H}}[F_{lsb||t-clear}]$ realizovaná $\mathcal{G}$ a $\mathcal{H}$ je bezpečná v zmysle RNM-CPA.*

Dôkaz. Nech $\mathcal{A}$ je útočník útočiaci na bezpečnosť kryptosystému $\mathrm{OAEP}^{\mathcal{G},\mathcal{H}}[F_{lsb||t-clear}]$ v zmysle RNM-CPA a nech $R_{\mathcal{A}}$ je polynomiálna binárna relácia pre dvojicu správ, ktorej platnosť chce útočník dosiahnuť. Dôkaz budeme konštruovať ako postupnosť hier podľa Shoupovej metodiky. Pre každý prechod medzi hrou charakterizovanou náhodnou veličinou $\mathrm{Exp}^{i}_{\mathcal{A},R_{\mathcal{A}},b}(k)$ a hrou charakterizovanou náhodnou veličinou $\mathrm{Exp}^{i+1}_{\mathcal{A},R_{\mathcal{A}},b}(k)$ pre $i = 0, 1, 2$ a fixný bit b ukážeme, že pravdepodobnosť úspechu útočníka $\mathcal{A}$ sa líši vždy nanajvýš o zanedbateľnú funkciu. V poslednej hre bude cieľový šifrový text nezávislý od príslušnej správy, pričom zároveň výstupy hier $\mathrm{Exp}^{3}_{\mathcal{A},R_{\mathcal{A}},b}(k)$ a $\mathrm{Exp}^{2}_{\mathcal{A},R_{\mathcal{A}},b}(k)$ budú výpočtovo nerozlíšiteľné. Tým dospejeme k záveru, že rovnako aj výstupy hier $\mathrm{Exp}^{3}_{\mathcal{A},R_{\mathcal{A}},b}(k)$ a $\mathrm{Exp}^{0}_{\mathcal{A},R_{\mathcal{A}},b}(k)$ musia byť výpočtovo nerozlíšiteľné.
Začneme charakterizáciou úvodnej hry Hra^0, ktorá popisuje reálny priebeh útoku.

Náhodná veličina $\mathrm{Exp}^{0}_{\mathcal{A},R_{\mathcal{A}},b}(k)$:
$K(1^k) = (pk, sk)$
Spočítaj šifrový text $X^*||\gamma^*||t^*$:
$M^* \leftarrow_{rnd} \mathcal{M}$
Ak $b = 0$ potom $M' = M^*$, inak $M' \leftarrow_{rnd} \mathcal{M}$
$r^* \leftarrow_{rnd} \{0,1\}^{k_0}$
$s^*_{k-k_0-k_1}||\gamma^* = \mathrm{G}(K_G, r^*) \oplus (M'||0^{k_1})$
$X^* = f(s^*_{k-k_0-k_1})$
$t^* = \mathrm{H}(K_H, s^*_{k-k_0-k_1}||\gamma^*) \oplus r^*$
$\mathcal{A}(X^*||\gamma^*||t^*, pk) = X||\gamma||t$
$D(sk, X||\gamma||t) = M$
Ak $(M^*, M) \in R_{\mathcal{A}}$ potom $\mathrm{Exp}^{0}_{\mathcal{A},R_{\mathcal{A}},b}(k) = 1$, inak $\mathrm{Exp}^{0}_{\mathcal{A},R_{\mathcal{A}},b}(k) = 0$

Pokračujme charakterizáciou nasledujúcej hry Hra^1.

Náhodná veličina $\mathrm{Exp}^1_{\mathcal{A},R_\mathcal{A},b}(k)$:
$K(1^k) = (pk, sk)$
Spočítaj šifrový text $X^*||\gamma^*||t^*$:
$M^* \leftarrow_{rnd} \mathcal{M}$
Ak $b = 0$ potom $M' = M^*$, inak $M' \leftarrow_{rnd} \mathcal{M}$
$r^* \leftarrow_{rnd} \{0,1\}^{k_0}$
$s^*_{k-k_0-k_1}||\gamma^* = \mathrm{G}(K_G, r^*) \oplus (M'||0^{k_1})$
$X^* = f(s^*_{k-k_0-k_1})$
$s'_{k-k_0-k_1}||\gamma' \leftarrow_{rnd} \{0,1\}^k$
$t^* = \mathrm{H}(K_H, s'_{k-k_0-k_1}||\gamma') \oplus r^*$
$\mathcal{A}(X^*||\gamma'||t^*, pk) = X||\gamma||t$
$D(sk, X||\gamma||t) = M$
Ak $(M^*, M) \in R_\mathcal{A}$ potom $\mathrm{Exp}^1_{\mathcal{A},R_\mathcal{A},b}(k) = 1$, inak $\mathrm{Exp}^1_{\mathcal{A},R_\mathcal{A},b}(k) = 0$

Rozdielom medzi hrou Hra^0 a hrou Hra^1 je zámena hodnoty $\mathrm{H}(K_H, s^*_{k-k_0-k_1}||\gamma^*)$ za $\mathrm{H}(K_H, s'_{k-k_0-k_1}||\gamma')$, pričom hodnotu $s'_{k-k_0-k_1}||\gamma'$ volíme náhodne a uniformne, teda táto hodnota je nezávislá na $s^*_{k-k_0-k_1}||\gamma^*$. Ukážeme, že vďaka predpokladu neohybnosti pseudonáhodného generátora $\mathcal{H}$ sú náhodné veličiny $\mathrm{Exp}^0_{\mathcal{A},R_\mathcal{A},b}(k)$ a $\mathrm{Exp}^1_{\mathcal{A},R_\mathcal{A},b}(k)$ výpočtovo nerozlíšiteľné. Uvažujme algoritmus $\mathcal{B}_b$ pre fixný bit $b \in \{0,1\}$, ktorý útočí na neohybnosť generátora $\mathcal{H}$ a istú polynomiálnu binárnu reláciu $R_\mathcal{B}$ založenú na relácii $R_\mathcal{A}$ (relácia $R_\mathcal{A}$ je podľa predpokladu stanovená ešte pred začiatkom útoku), ktorá je podprogramom algoritmu $\mathcal{B}_b$. Relácia $R_\mathcal{B}$ je definovaná nasledovne. Súčasťou popisu $R_\mathcal{B}$ je aj vygenerovaná dvojica (K_G, K_G^{-1}) (pripomeňme, že podľa predpokladu je $\mathcal{G}$ bezkolízny pseudonáhodný generátor s padacími vrátkami) a popis relácie $R_\mathcal{A}$. Na vstupe $(s^*_{k-k_0-k_1}||\gamma^*, s_{k-k_0-k_1}||\gamma)$ relácia $R_\mathcal{B}$ najskôr využitím padacích vrátok K_G^{-1} a hodnôt γ^* a γ spočíta odpovedajúce hodnoty r^* a r. Následne relácia $R_\mathcal{B}$ spočíta $M^*||0^{k_1} = (s^*_{k-k_0-k_1}||\gamma^*) \oplus \mathrm{G}(K_G, r^*)$ a tiež $M||z = (s_{k-k_0-k_1}||\gamma) \oplus \mathrm{G}(K_G, r)$. Relácia $R_\mathcal{B}$ platí (tj. jej výstupom je 1), práve ak je $z = 0^{k_1}$ a zároveň $(M^*, M) \in R_\mathcal{A}$. Algoritmus $\mathcal{B}_b$ obdrží na vstupe usporiadanú štvoricu $(K_G^{-1}, f, f(s^*_{k-k_0-k_1}||\gamma^*), y^*)$, kde buď $y^* = \mathrm{H}(K_H, s^*_{k-k_0-k_1}||\gamma^*)$ alebo $y^* = \mathrm{H}(K_H, s'_{k-k_0-k_1}||\gamma')$ pre $s'_{k-k_0-k_1}||\gamma'$ nezávislé na $s^*_{k-k_0-k_1}||\gamma^*$. Následne $\mathcal{B}_b$ spustí simuláciu útočníka $\mathcal{A}$, ktorému ako verejný kľúč predá trojicu $pk = (f, K_G, K_H)$, pričom padacie vrátka K_G^{-1} (súčasť popisu relácie $R_\mathcal{B}$) si uloží. Pre generáciu cieľového šifrového textu použije algoritmus $\mathcal{B}_b$ hodnotu r^*, ktorú spočíta práve vďaka znalosti padacích vrátok K_G^{-1} a hodnoty γ^* (tá je súčasťou jeho vstupu). Algoritmus $\mathcal{B}_b$ ďalej spočíta $t^* = y^* \oplus r^*$ a ako cieľový šifrový text vydá útočníkovi $\mathcal{A}$ hodnotu $f(s^*_{k-k_0-k_1})||\gamma^*||t^*$. Útočník $\mathcal{A}$ následne ako svoj výstup vráti nejaký šifrový text $f(s_{k-k_0-k_1})||\gamma||t$. Ak je tento šifrový text platný, algoritmus $\mathcal{B}_b$ vďaka znalosti padacích vrátok K_G^{-1} vypočíta pomocou hodnoty γ odpovedajúcu hodnotu r. Nakoniec, algoritmus $\mathcal{B}_b$ vráti ako svoj výstup dvojicu $(f(s_{k-k_0-k_1})||\gamma, t \oplus r)$. Všimnime si, že vzťah $r^* \neq r$ implikuje $\gamma = \mathrm{lsb}_{k_1}(\mathrm{G}(K_G, r)) \neq \mathrm{lsb}_{k_1}(\mathrm{G}(K_G, r^*)) = \gamma^*$ s pravdepodobnosťou líšiacou sa od 1 nanajvýš o zanedbateľnú funkciu. V opačnom prípade (tj. ak by platilo $\gamma = \gamma^*$) by totiž hodnoty r a r^*, ktoré algoritmus $\mathcal{B}_b$

získal, znamenali spor s bezkolíznosťou generátora $\mathcal{G}$. Tým sme dospeli k záveru, že musia byť rozdielne aj hodnoty $s^*_{k-k_0-k_1}||\gamma^*$ a $s_{k-k_0-k_1}||\gamma$, ako je požadované pre úspech útoku proti slabej neohybnosti daného kryptosystému. Pokiaľ platí $r^* = r$, potom musí byť $s^*_{k-k_0-k_1} \neq s_{k-k_0-k_1}$, inak by bol šifrový text vygenerovaný útočníkom zhodný s cieľovým šifrovým textom a útok proti slabej neohybnosti daného kryptosystému by bol neúspešný. Navyše, ak je šifrový text vygenerovaný útočníkom platný, potom je výstupom algoritmu $\mathcal{B}_b$ platná dvojica $(f(s_{k-k_0-k_1})||\gamma, \mathrm{H}(K_H, s_{k-k_0-k_1}||\gamma))$, ktorú $\mathcal{B}_b$ získa z výstupu $\mathcal{A}$. Čiže pravdepodobnosť, že pre dané $y^* = \mathrm{H}(K_H, s^*_{k-k_0-k_1}||\gamma^*)$ je výstupom algoritmu $\mathcal{B}_b$ hodnota 1, sa od pravdepodobnosti, že výsledkom hry Hra^0 je hodnota $\mathrm{Exp}^0_{\mathcal{A},R_\mathcal{A},b}(k) = 1$, líši nanajvýš o funkciu zanedbateľnú v k. Podobne pre hodnotu $y^* = \mathrm{H}(K_H, s'_{k-k_0-k_1}||\gamma')$ je výstupom algoritmu $\mathcal{B}_b$ hodnota 1 s pravdepodobnosťou, ktorá sa od pravdepodobnosti, že výsledkom hry Hra^1 je hodnota $\mathrm{Exp}^1_{\mathcal{A},R_\mathcal{A},b}(k) = 1$, líši nanajvýš o zanedbateľnú funkciu. Preto sú výstupy oboch hier Hra^1 a Hra^0 výpočtovo nerozlíšiteľné.
Pokračujme charakterizáciou ďalšej hry Hra^2.

Náhodná veličina $\mathrm{Exp}^2_{\mathcal{A},R_\mathcal{A},b}(k)$:
- $K(1^k) = (pk, sk)$
- Spočítaj šifrový text $X^*||\gamma^*||t^*$:
  - $M^* \leftarrow_{rnd} \mathcal{M}$
  - Ak $b = 0$ potom $M' = M^*$, inak $M' \leftarrow_{rnd} \mathcal{M}$
  - $r^* \leftarrow_{rnd} \{0,1\}^{k_0}$
  - $s^*_{k-k_0-k_1}||\gamma^* = \mathrm{G}(K_G, r^*) \oplus (M'||0^{k_1})$
  - $X^* = f(s^*_{k-k_0-k_1})$
  - $\gamma' \leftarrow_{rnd} \{0,1\}^{k_1};\ u' \leftarrow_{rnd} \{0,1\}^{k_0}$
  - $t^* = u' \oplus r^*$
- $\mathcal{A}(X^*||\gamma'||t^*, pk) = X||\gamma||t$
- $D(sk, X||\gamma||t) = M$
- Ak $(M^*, M) \in R_\mathcal{A}$ potom $\mathrm{Exp}^2_{\mathcal{A},R_\mathcal{A},b}(k) = 1$, inak $\mathrm{Exp}^2_{\mathcal{A},R_\mathcal{A},b}(k) = 0$

Rozdielom medzi hrou Hra^1 a hrou Hra^2 je zámena hodnoty $\mathrm{H}(K_H, s'_{k-k_0-k_1}||\gamma')$ za hodnotu u', ktorá je volená náhodne a uniformne. Opäť chceme ukázať, že rozdelenie náhodných veličín $\mathrm{Exp}^1_{\mathcal{A},R_\mathcal{A},b}(k)$ a $\mathrm{Exp}^2_{\mathcal{A},R_\mathcal{A},b}(k)$ charakterizujúcich jednotlivé hry sa touto zámenou výrazne nezmení. Uvažujme teda pre fixný bit $b \in \{0,1\}$ pravdepodobnostný polynomiálny algoritmus $\mathcal{D}_b$, ktorý je rozlišovačom pseudonáhodnosti generátora $\mathcal{H}$ vzhľadom k $\mathrm{hint}(s'_{k-k_0-k_1}||\gamma') = (f, f(s_{k-k_0-k_1}||\gamma)$. Algoritmus $\mathcal{D}_b$ dostane na vstupe trojicu (K_H, y', γ'), kde hodnota γ' je náhodná a pre hodnotu y' platí buď $y' = \mathrm{H}(K_H, s'_{k-k_0-k_1}||\gamma')$ pre náhodné $s'_{k-k_0-k_1}$ alebo $y' = u'$ pre náhodné u'. Algoritmus $\mathcal{D}_b$ najskôr vygeneruje dvojicu $(f, f^{-1}) \leftarrow_{rnd} F(1^k)$ a dvojicu $(K_G, K_G^{-1}) = \mathrm{KGenG}(1^k)$ a následne spustí simuláciu útočníka $\mathcal{A}$. Aby rozlišovač $\mathcal{D}_b$ vygeneroval pre útočníka $\mathcal{A}$ cieľový šifrový text, zvolí $M^* \leftarrow_{rnd} \mathcal{M}$ ($\mathcal{M}$ je podľa predpokladu uniformné rozdelenie) a pre $b = 0$ položí $M' = M^*$, prípadne pre $b = 1$ vygeneruje nové $M' \leftarrow_{rnd} \mathcal{M}$. Algoritmus $\mathcal{D}_b$ následne spočíta príslušnú hodnotu r^* vďaka znalosti hodnôt K_G^{-1} a γ' pomocou vzťahu $s^*_{k-k_0-k_1}||\gamma' = \mathrm{G}(K_G, r^*) \oplus (M'||0^{k_1})$. Ako cieľový šifrový text vydá útočníkovi $\mathcal{A}$ hodnotu $f(s^*_{k-k_0-k_1})||\gamma'||(y' \oplus r^*)$. Útočník ako svoj výstup vráti rozlišovaču

$\mathcal{D}_b$ nejaký šifrový text $f(s_{k-k_0-k_1})||\gamma||t$. Nakoniec, rozlišovač $\mathcal{D}_b$ dešifruje tento šifrový text pomocou znalosti f^{-1}, čím obdrží nejakú správu M a ako svoj výstup vráti 1, práve ak platí $(M^*, M) \in R_{\mathcal{A}}$. Čiže pravdepodobnosť, že pre dané $y' = \mathrm{H}(K_H, s'_{k-k_0-k_1}||\gamma')$ vráti rozlišovač $\mathcal{D}_b$ ako svoj výstup hodnotu 1 je rovnaká ako pravdepodobnosť, že výsledkom hry Hra^1 je hodnota $\mathrm{Exp}^1_{\mathcal{A},R_{\mathcal{A}},b}(k) = 1$. Podobne pre náhodné $y' = u'$ vráti rozlišovač $\mathcal{D}_b$ ako svoj výstup hodnotu 1 s pravdepodobnosťou, ktorá je rovná pravdepodobnosti, že výsledkom hry Hra^2 je hodnota $\mathrm{Exp}^2_{\mathcal{A},R_{\mathcal{A}},b}(k) = 1$. Preto sú výstupy oboch hier Hra^2 a Hra^1 výpočtovo nerozlíšiteľné.
Pokračujme charakterizáciou poslednej hry Hra^3.

Náhodná veličina $\mathrm{Exp}^3_{\mathcal{A},R_{\mathcal{A}},b}(k)$:
$K(1^k) = (pk, sk)$
Spočítaj šifrový text $X^*||\gamma^*||t^*$:
$M^* \leftarrow_{rnd} \mathcal{M}$
Ak $b = 0$ potom $M' = M^*$, inak $M' \leftarrow_{rnd} \mathcal{M}$
$r^* \leftarrow_{rnd} \{0,1\}^{k_0}$
$v^* \leftarrow_{rnd} \{0,1\}^{k-k_0}$
$s^*_{k-k_0-k_1}||\gamma^* = v^* \oplus (M'||0^{k_1})$
$X^* = f(s^*_{k-k_0-k_1})$
$\gamma' \leftarrow_{rnd} \{0,1\}^{k_1}$; $u' \leftarrow_{rnd} \{0,1\}^{k_0}$
$t^* = u' \oplus r^*$
$\mathcal{A}(X^*||\gamma'||t^*, pk) = X||\gamma||t$
$D(sk, X||\gamma||t) = M$
Ak $(M^*, M) \in R_{\mathcal{A}}$ potom $\mathrm{Exp}^3_{\mathcal{A},R_{\mathcal{A}},b}(k) = 1$, inak $\mathrm{Exp}^3_{\mathcal{A},R_{\mathcal{A}},b}(k) = 0$

Nerozlíšiteľnosť hier Hra^2 a hrou Hra^3 je možné ukázať analogickým postupom ako v predchádzajúcom prípade pomocou rozlišovača pseudonáhodnosti, tentokrát však pre generátor $\mathcal{G}$. Konštrukcia tohoto rozlišovača je podobná konštrukcii rozlišovača $\mathcal{D}_b$. Jediným rozdielom je, že hodnotu $t^* = u' \oplus r^*$ vymeníme za hodnotu $t^* = u'$. Toto je ale iba technický rozdiel, lebo vzhľadom k tomu, že hodnota u' je volená náhodne a uniformne, rozdelenie výslednej hodnoty t^* sa nemení. Výstupy oboch hier Hra^3 a Hra^2 sú teda opäť výpočtovo nerozlíšiteľné. Preto aj výstupy hier Hra^3 a Hra^0 (situácia pri reálnom útoku proti slabej neohybnosti kryptosystému) sú výpočtovo nerozlíšiteľné. V poslednej hre Hra^3 sú však už výstupy oboch pseudonáhodných generátorov $\mathcal{G}$ a $\mathcal{H}$ nahradené náhodnými hodnotami a útočník dostáva v každom prípade ako vstup cieľový šifrový text, ktorý je nezávislý na pôvodnej správe M^*. Tým sme dokázali bezpečnosť plnej inštancializácie $\mathrm{OAEP}^{\mathcal{G},\mathcal{H}}[F_{lsb||t-clear}]$ realizovanej $\mathcal{G}$ a $\mathcal{H}$ v zmysle RNM-CPA.

□

5 Záver

V tejto práci sme sa zoznámili s niekoľkými podstatnými výsledkami týkajúcimi sa preukázateľnej bezpečnosti na príklade jedného z najpoužívanejších kryptosystémov súčasnosti – OAEP. Cieľom bolo predstaviť súčasný stav tejto oblasti a ilustrovať možnosti preukázateľnej bezpečnosti v dvoch rovinách, a síce využitím:

- modelu náhodného orákula,
- štandardného modelu.

Nasledujúce odstavce venujeme niekoľkým komentárom k súčasnému stavu preukázateľnej bezpečnosti z pohľadu obsahu tejto práce, pričom sa budeme sústrediť na porovnanie oboch študovaných prístupov. Výborným podkladom k tomuto účelu bola publikácia [**11**] od dvojice autorov Katz a Lindell.

Model náhodného orákula sa dnes teší značnej popularite a je v praxi hojne používaný. Poskytuje akýsi kompromis medzi formálnym matematickým dôkazom bezpečnosti na jednej strane a vôbec žiadnym dôkazom na strane druhej. Dokazovanie bezpečnosti je totiž v modele náhodného orákula pojaté idealizovane a nezachycuje presný odraz reality. Avšak napriek tomu je pomocou dôkazu bezpečnosti v modele náhodného orákula možné demonštrovať rozumnosť konštrukcie daného kryptosystému, ktorá sa často stáva najsilnejším dostupným argumentom k vybudovaniu dôvery v praktickú bezpečnosť tohoto kryptosystému – a podobne často je ako taký argument odbornou komunitou aj akceptovaná.

Ako sme mali možnosť v tejto práci vidieť, model náhodného orákula poskytuje formálnu metodiku, ktorú je možné použiť k návrhu a validácii bezpečnosti kryptosystému nasledujúcim spôsobom:

- Kryptosystém je navrhnutý a pomocou modelu náhodného orákula je dokázaná jeho bezpečnosť. Daný dôkaz teda platí za predpokladu, že v realite existuje náhodné orákulum. Okrem tohto predpokladu môžu byť pri dokazovaní použité aj štandardné kryptografické predpoklady týkajúce sa výpočtovej nedosažiteľnosti niektorých dobre známych problémov.
- Pri praktickej implementácii daného kryptosystému žiadne náhodné orákulum nemáme k dispozícii. Náhodné orákulá využité k dokázaniu bezpečnosti tohoto kryptosystému teda musíme nahradiť (tj. inštancializovať) nejakou vhodnou kryptografickou hashovacou funkciou (napríklad SHA-1). To znamená, že v každom kroku, kde sa pri výpočtoch súvisiacich s kryptosystémom (šifrovanie, dešifrovanie, útok) nejaká entita pred inštancializáciou dotazovala náhodného orákula, je po inštancializácii táto entita namiesto dotazovania schopná odpoveď na dotaz vypočítať sama pomocou vlastných

zdrojov a všeobecne známeho popisu použitej kryptografickej hashovacej funkcie.

Pre zachovanie platnosti dôkazu bezpečnosti v modele náhodného orákula v praxi teda ostáva dúfať, že kryptografická hashovacia funkcia z predchádzajúcej odrážky svojimi vlastnosťami dostatočne dobre emuluje náhodné orákulum. Slovo dúfať je použité zámerne, keďže v súčasnosti neexistuje teoretické opodstatnenie zachovania bezpečnostných vlasností pri inštancializácii. Dokonca sú známe konštrukcie kryptosystémov, ktoré sú bezpečné v modele náhodného orákula, no súčasne ľubovoľná inštancializácia náhodného orákula vedie k strate dokázanej bezpečnosti (viď prácu [**7**]). Navyše ani nie je jasné, čo presne musí kryptografická hashovacia funkcia splňovať, aby v praxi dobre emulovala náhodné orákulum a rovnako ani nie je známe, či je vôbec také podmienky možné definovať.

Poďme si zhrnúť, v čom spočíva hlavný rozdiel medzi dôkazmi bezpečnosti v modele náhodného orákula a v štandardnom modele. Nech teda Π označuje kryptosystém, ktorého bezpečnosť dokazujeme, $\mathcal{A}$ označuje útočníka, ktorý na tento kryptosystém útočí, $\mathrm{Exp}_{\mathcal{A},\Pi}$ označuje scenár útoku, voči ktorému chceme bezpečnosť kryptosystému dokázať a γ označuje požadovanú pravdepodobnosť, že útočník bude pri tomto útoku úspešný (napr. pri útoku s voľbou šifrového textu je $\gamma = 1/2$, pretože požadujeme, aby útočník nebol schopný rozoznať cieľový šifrový text vygenerovaný z jednej z dvoch ním určených správ). V štandardnom modele je schéma Π bezpečná proti útoku $\mathrm{Exp}_{\mathcal{A},\Pi}$, ak pre ľubovoľného pravdepodobnostného polynomiálneho útočníka a všetky dostatočne veľké $k \in \mathbb{N}$ platí

$$\Pr[\mathrm{Exp}_{\mathcal{A},\Pi}(k) = 1] \leq \gamma + \delta(k),$$

kde $\delta(k)$ je nejaká zanedbateľná funkcia a pravdepodobnosť je počítaná cez náhodné voľby všetkých entít zúčastnených na behu kryptosystému Π, vrátane útočníka $\mathcal{A}$. Naproti tomu je situácia v modele náhodného orákula nasledujúca. Označme ako H náhodné orákulum zakomponované do kryptosystému Π^H. Potom je schéma Π^H bezpečná proti útoku $\mathrm{Exp}_{\mathcal{A}^H,\Pi^H}$, ak pre ľubovoľného pravdepodobnostného polynomiálneho útočníka a všetky dostatočne veľké $k \in \mathbb{N}$ platí

$$\Pr[\mathrm{Exp}_{\mathcal{A}^H,\Pi^H}(k) = 1] \leq \gamma + \delta(k),$$

kde $\delta(k)$ je nejaká zanedbateľná funkcia, pričom pravdepodobnosť je počítaná cez náhodné voľby všetkých entít zúčastnených na behu kryptosystému Π, vrátane útočníka $\mathcal{A}$ a navyše cez náhodné voľby náhodného orákula H. Pri reálnom nasadení kryptosystému Π však musíme vybrať nejaké konkrétne H. Bezpečnosť Π ale pritom nie je garantovaná pre žiadne konkrétne Π, je garantovaná iba cez náhodné voľby H. Preto pri inštancializácii H napríklad hashovacou funkciou SHA-1 môže nastať, že práve táto inštancializácia nezachová bezpečnostné vlastnosti garantované dôkazom v modele náhodného orákula. Poznamenajme ešte, že model náhodného orákula získal svoj názov práve vďaka tejto odlišnosti od štandardného modelu.

Predchádzajúce úvahy nás privádzajú k niekoľkým zásadným otázkam:

- Čo vlastne dôkaz bezpečnosti v modele náhodného orákula znamená v realite?

- Je dôkaz bezpečnosti v modele náhodného orákula principiálne odlišný od dôkazu bezpečnosti v štandardnom modele?

Uvedené otázky momentálne nemajú jednoznačnú odpoveď, ktorá by bola všeobecne akceptovaná. Naopak, sú predmetom aktívneho výskumu dnešnej kryptografie a tiež množstva polemík v rámci odbornej komunity. Pokúsime sa na tomto mieste zhrnúť súčasné argumenty pre a proti modelu náhodného orákula.

Kritika modelu náhodného orákula. Základným nedostatkom modelu náhodného orákula je, ako sme poznamenali už vyššie, neexistencia teoretického opodstatnenia pre podporu domnienky, že dôkaz bezpečnosti daného kryptosystému v modele náhodného orákula implikuje čokoľvek o bezpečnosti tohto kryptosystému v reálnom svete (po inštancializácii náhodného orákula ľubovoľnou kryptografickou hashovacou funkciou). Tento fakt nie je len prostou teoretickou neistotou, pretože žiadna kryptografická hashovacia funkcia v realite nikdy nemôže dosiahnuť vlastností náhodného orákula. Pre ilustráciu si stačí uvedomiť, že v modele náhodného orákula je odpoveďou na dotaz x hodnota $H(x)$, ktorá je pre okolitý svet skutočne náhodná – pokiaľ dané x ešte nebolo náhodného orákula H explicitne dotázané, má o hodnote $H(x)$ akákoľvek entita nulovú informáciu. Toto tvrdenie platí dokonca aj pre entitu s neobmedzenými výpočtovými zdrojmi (stačí iba polynomiálne obmedziť prípustný počet dotazov), čo v reálnom svete nemá obdobu. Entita s neobmedzenými výpočtovými zdrojmi totiž v praxi dokáže sama každú konkrétnu hashovaciu funkciu nielen vyčísliť (k danému x určiť odpovedajúcu hodnotu $H(x)$), ale navyše aj invertovať (k danému $H(x)$ určiť odpovedajúcu hodnotu x). Avšak aj pre entitu s polynomiálne obmedzenými zdrojmi platí, že akonáhle je náhodné orákulum inštancializované akoukoľvek kryptografickou hashovacou funkciou, je pre každé x z jej definičného oboru príslušná hodnota $H(x)$ okamžite definovaná a ktokoľvek so znalosťou x ju môže vypočítať. Ďalšou zásadnou deviáciou modelu náhodného orákula od reality sú spôsoby jeho využitia v dôkazoch bezpečnosti. Konkrétne napríklad algoritmus slúžiaci k redukcii je schopný všetky dotazy útočníka na náhodné orákulum sledovať, čo sa pre niektoré dôkazy bezpečnosti v modele náhodného orákula ukazuje byť vitálnym predpokladom (viď dôkaz v oddieli 2.2.2). Opäť, pri inštancializácii náhodného orákula kryptografickou hashovacou funkciou toto nie je možné, pretože popis tejto hashovacej funkcie je útočníkovi (a všetkým ostatným) dopredu známy a tak je útočník schopný vypočítať ku každému x príslušné $H(x)$ sám bez realizovania jediného explicitného dotazu. Iným príkladom využitia idealizovaných vlastností modelu náhodného orákula v dôkazoch bezpečnosti je možnosť algoritmu slúžiacemu k redukcii definovať pre každý dotaz x príslušnú odpoveď úplne ľubovoľne (pokiaľ sa táto odpoveď javí dotazovacej entite ako náhodná a konzistentná pri opakovaných dotazoch), čo býva taktiež stavebným kameňom dôkazov v modele náhodného orákula (opäť viď dôkaz v oddieli 2.2.2) a čo taktiež nemá existujúcu paralelu v reálnom svete. Odhliadnuc od uvedených teoretických problémov, model náhodného orákula trpí i výraznými praktickými nedostatkami. Ako uvádzame vyššie, momentálne nie je známe, čo znamená pre konkrétnu hashovaciu funkciu dostatočne dobrá inštancializácia náhodného orákula. Predstavme si, že by sme náhodné orákulum chceli inštancializovať napríklad hashovacou funkciou SHA-1. Môže nastať prípad, že pre nejaký konkrétny uvažovaný kryptosystém

je inštancializácia pomocou SHA-1 dostatočne bezpečnou voľbou. Je však minimálne odvážnym predpokladať, že taká inštancializácia bude dostatočne bezpečnou voľbou pre každý kryptosystém s dôkazom bezpečnosti v modele náhodného orákula, pretože SHA-1 nie je náhodným orákulom (v skutočnosti ním nie je ani žiadna iná hashovacia funkcia Merkle-Damgardovho typu). Predpoklad tohto charakteru (tj. SHA-1 sa správa ako náhodné orákulum) je výrazne odlišný od predpokladov typu "SHA-1 je bezkolízna hashovacia funkcia" alebo "AES je pseudonáhodná funkcia". Problémom je fakt, že v posledne menovaných prípadoch máme k dispozícii použiteľné formálne definície, voči ktorým sú dané tvrdenia verifikovateľné, pričom pre aproximáciu náhodného orákula reálnou hashovacou funkciou žiadna podobná definícia neexistuje. Kvôli uvedeným nedostatkom je použitie modelu náhodného orákula kvalitatívnym rozdielom oproti iným predpokladom, ktoré sú bežne využívané pri dokazovaní bezpečnosti kryptosystémov v štandardnom modele (napr. zavedenie nového kryptografického predpokladu o výpočtovej nedosažiteľnosti nejakého problému). Preto sú dôkazy bezpečnosti v modele náhodného orákula menej žiadané a majú principiálne menšiu váhu než dôkazy v štandardnom modele.

Podpora modelu náhodného orákula. Po prečítaní predchádzajúceho odstavca si iste položíme logickú otázku – keď má model náhodného orákula také nedostatky, prečo by sme ho vôbec mali používať? Alebo ešte inak – prečo má model náhodného orákula tak výrazný vplyv na vývoj modernej kryptografie a je v praxi tak široko používaný? Odpovedí je niekoľko. Ako sme v tejto práci mali možnosť vidieť, model náhodného orákula umožňuje navrhovať podstatne efektívnejšie kryptosystémy s dôkazom bezpečnosti ako štandardný model (viď oddiel 4.2). V súčasnosti totiž existuje len niekoľko málo prakticky použiteľných asymetrických kryptosystémov s dôkazom bezpečnosti v štandardnom modele. Nie je napríklad známy žiaden kryptosystém založený na šifrovacej transformácii RSA, ktorý by mal dostatočne silný dôkaz bezpečnosti v štandardnom modele a zároveň by bol dostatočne efektívny pre praktické použitie (množstvo ľudí v realite totiž radšej než pomalý kryptosystém nebude používať vôbec žiadny kryptosystém). Situáciu dobre ilustruje oddiel 4.2, kde sme mali možnosť vidieť dôkaz bezpečnosti v štandardnom modele. Tento dôkaz je však platný iba pre prakticky nepoužívanú variantu OAEP (ktorá bola vytvorená čisto pre teoretické účely tohto dôkazu) a i tak je jeho váha prislabá (RNM-CPA poskytuje absolútne nedostačujúcu garanciu bezpečnosti pre praktické použitie). Navyše, ako sme si ukázali, solídnu mieru bezpečnosti OAEP, ktoré je momentálne najpoužívanejším asymetrickým kryptosystémom súčasnosti, v štandardnom modele dokonca ani principiálne nie je možné dokázať (viď oddiel 4.1). Vďaka prípadom podobným OAEP sa nám teda postupne odkrýva výrazne obmedzená hranica možností, na ktorú pri dokazovaní bezpečnosti v štandardnom modele neustále narážame. Oproti tomu stojí množstvu hojne využívaných asymetrických kryptosystémov s dôkazom bezpečnosti v modele náhodného orákula, vrátane práve OAEP (viď oddiel 2.3). Okrem toho, existujúci dôkaz bezpečnosti v modele náhodného orákula sa postupne začína považovať za dôležitú (ak nie nutnú) súčasť kryptosystémov posudzovaných ako budúce štandardy. Model náhodného orákula tiež podporil dôveru v niektoré efektívne kryptosystémy, ktoré sa v praxi dlhšiu dobu považujú za bezpečné i bez existujúceho dôkazu bezpečnosti v štandardnom modele

(ako príklad môžeme opäť uviesť kryptosystém OAEP). Tým sa do veľkej miery zaslúžil o masívnejšiu penetráciu kryptografie ako takej do každodennej ľudskej komunikácie. Základným argumentom pre podporu modelu náhodného orákula je teda pozorovanie, že dôkazy bezpečnosti v modele náhodného orákula sú podstatne lepšie než vôbec žiadne dôkazy bezpečnosti, a to minimálne z nasledujúcich dvoch dôvodov:

- Dôkaz bezpečnosti daného kryptosystému v modele náhodného orákula svedčí o rozumnosti návrhu tohto kryptosystému v zmysle, že jediné jeho možné slabiny môžu vyvstať z inštancializácie náhodného orákula pomocou nejakej konkrétnej kryptografickej hashovacej funkcie. Inak povedané, dôkaz v modele náhodného orákula hovorí, že jedinou cestou ako v praxi zlomiť bezpečnosť daného kryptosystému, je nejakým spôsobom k tomu využiť vlastnosti príslušnej hashovacej funkcie, resp. možnú zmenu bezpečnostných vlastností celého kryptosystému vyvolanú inštancializáciou. Preto ak je použitá hashovacia funkcia dostatočne kvalitná (nanešťastie ale nevieme, čo presne to znamená), môžeme mať v praktickú bezpečnosť tohto kryptosystém dôveru. Navyše, ak je náhodou príslušná kryptografická hashovacia funkcia v realite zlomená, môžeme ju nahradiť inou hashovacou funkciou, ktorá nemá zistené nedostatky.
- Existuje iba málo známych reálne nebezpečných útokov proti prakticky používaným kryptosystémom, ktoré sú bezpečné v modele náhodného orákula. Tento argument má síce špekulatívny charakter, ale napriek tomu je značne dôležitý práve kvôli svojej praktickej merateľnosti.

Okrem uvedených špeciálnych problémov vyplývajúcich z rozdielneho prístupu k dokazovaniu bezpečnosti, sme sa rovnako mali možnosť zoznámiť s radou ďalších problémov všeobecného charakteru. Za všetky spomeňme slabú výpovednú hodnotu dôkazov bezpečnosti s nepriliehavou redukciou (viď oddiel 3.1), bezpečnostné zraniteľnosti vznikajúce pri nedôslednej implementácii inak bezpečných kryptosystémov (viď oddiely 3.2 a 3.3), ale tiež napríklad omnoho vážnejší problém, akým je nedostatočná spätná väzba odbornej komunity k publikovaným dôkazom bezpečnosti. Tu si stačí uvedomiť, že kryptosystém OAEP bol na základe článku autorov Bellare a Rogaway z roku 1994 všeobecne považovaný za bezpečný (a s týmto vedomím i široko využívaný), až kým sa neobjavila Shoupova námietka, ktorá jeho korektnosť vyvrátila. To sa ale stalo až v roku 2001, teda po uplynutí celých siedmich rokov...

V každom prípade, preukázateľná bezpečnosť ako matematický obor aktuálne čelí mnohým fundamentálnym otázkam, ktoré zatiaľ ostávajú nezodpovedané. Ich zodpovedanie predstavuje veľkú výzvu pre kryptológov súčasnosti vzhľadom k určujúcemu potenciálu hľadaných odpovedí pre budúci vývoj chápania pojmu bezpečnosť.

Literatúra

[1] M. Bellare, A. Desai, D. Pointcheval, P. Rogaway: *Relations among Notions of Security for Public-Key Encryption Schemes.* Advances in Cryptology – CRYPTO 98, 26–45, 1998.

[2] M. Bellare, P. Rogaway: *Optimal Asymmetric Encryption – How to Encrypt with RSA.* Advances in Cryptology – EUROCRYPT 94, 92–111, 1994.

[3] D. Bleichenbacher: *Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1.* Advances in Cryptology – CRYPTO 98, 1–12, 1998.

[4] A. Boldyreva, M. Fischlin: *On the Security of OAEP.* Advances in Cryptology – Asiacrypt 2006 Proceedings, 210-225, 2006.

[5] D. Boneh: *Simplifed OAEP for the RSA and Rabin functions.* Advances in Cryptology – CRYPTO 01, 275–291, 2001.

[6] D. Brown: *A Weak-Randomizer Attack on RSA-OAEP with* $e = 3$. http://eprint.iacr.org/2005/189.pdf, 2005.

[7] R. Canetti, O. Goldreich, S. Halevi: *The Random Oracle Methodology, Revisited.* Proceedings of the 30th ACM Annual Symposium on Theory of Computing, 209–218, 1998.

[8] D. Coppersmith: *Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities.* Journal of Cryptology, 233–260, 1997.

[9] E. Fujisaki, T. Okamoto, D. Pointcheval, J. Stern: *RSA-OAEP Is Secure under the RSA Assumption.* Journal of Cryptology, 81–104, 2004.

[10] R. Gennaro, L. Trevisan: *Lower Bounds on the Efficiency of Generic Cryptographic Constructions.* Proceedings of FOCS'00, 305–313, 2000.

[11] J. Katz, Y. Lindell: *Introduction To Modern Cryptography.* Chapman & Hall, 2007.

[12] E. Kiltz, K. Pietrzak: *On the Security of Padding-Based Encryption Schemes.* Proceedings of IACR EUROCRYPT 2009, 389–406, 2009.

[13] N. Koblitz, A. Menezes: *Another Look at "Provable Security".* Cryptology ePrint Archive, Report 2004/152, 2004.

[14] J. Manger: *A Chosen Ciphertext Attack on RSA Optimal Asymmetric Encryption Padding (OAEP) as Standardized in PKCS #1 v2.0.*

[15] A. Menezes, P. van Oorschot, S. Vanstone: *Handbook of Applied Cryptography.* CRC Press, ISBN 0-8493-8523-7, 2006. Dostupné na http://www.cacr.math.uwaterloo.ca/hac

[16] D. Pointcheval: *Contemporary Cryptology – Provable Security for Public Key Schemes.* Advanced Courses CRM Barcelona, 133–189, 2005.

[17] M. Rabin: *Digitalized Signatures and Public-Key Functions as Intractable as Factorization.* MIT Laboratory for Computer Science, 1979.

[18] R. Rivest, A. Shamir, L. Adleman: *A method for obtaining digital signatures and public-key cryptosystems.* Communications of the ACM, 120–126, 1978.

[19] V. Shoup: *OAEP Reconsidered.* Journal of Cryptology, 239–259, 2001.

[20] V. Shoup: *Sequences of Games: A Tool for Taming Complexity in Security Proofs.* http://www.shoup.net/papers/games.pdf, 2005.