FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
UNIVERZITA KOMENSKÉHO
Katedra informatiky

Podpisové schémy umožňujúce odstraňovanie častí dokumentu

Diplomová práca

Autor : Martin Dzurenko
Vedúci: RNDr. Martin Stanek, PhD.

Bratislava, 2005

Poďakovanie

Touto cestou vyslovujem poďakovanie RNDr. Martinovi Stanekovi, PhD. za poskytnutie literatúry, odborné vedenie, cenné rady a pripomienky pri vypracovaní mojej diplomovej práce.

Čestne vyhlasujem, že diplomovú prácu som vypracoval samostatne a že som uviedol všetku použitú literatúru.

.....

V Bratislave, apríl 2005 Martin Dzurenko

Obsah

Kapitola 1

Úvod

Cieľom tejto diplomovej práce je preskúmať možnosti vytvárania, transformácie a overovania podpisov s možnosťou odstraňovania častí podpísaného dokumentu a navrhnúť modifikácie už existujúcich schém. Týmto požiadavkám zodpovedá aj členenie práce na jednotlivé kapitoly.

V druhej kapitole sa budeme zaoberať návrhom týchto schém. Pre zjednodušenie budeme podpisové schémy umožňujúce odstraňovanie častí podpísaného dokumentu označovať skratkou CES (z angl. Content Extraction Signatures). Rozoberieme požiadavky, ktoré viedli k vzniku takýchto schém a motivačný príklad. Zavedieme terminológiu potrebnú na špecifikáciu funkčných a bezpečnostných požiadaviek na takéto schémy a takisto na samotnú definíciu schém CES. Načrtneme možné aplikácie schém a nakoniec uvedieme jednu konkrétnu implementáciu, a to schému RSAProd [3].

V tretej kapitole popíšeme jeden z kľúčových konceptov pri tvorbe takýchto podpisov – prístupovú štruktúru. Jej úlohou bude stanoviť pravidlá, ktorými sa bude musieť riadiť každý, kto chce modifikovať pôvodný dokument a s ním asociovaný podpis. Tieto pravidlá stanoví autor dokumentu. Popíšeme a porovnáme rôzne typy prístupových štruktúr. Navyše definujeme novú grafovú reprezentáciu prístupovej štruktúry.

V štvrtej kapitole navrhneme novú stromovú prístupovú štruktúru kombinujúcu výhody existujúcich štruktúr. Uvedieme jej implementáciu, jej výhody a nevýhody. Nakoniec uvedieme príklady konkrétneho použitia tejto prístupovej štruktúry.

V piatej kapitole podrobne popíšeme modifikáciu schémy RSAProd, ktorá využíva stromovú prístupovú štruktúru zavedenú v kapitole 4. Uvedieme samotné algoritmy, príklady ilustrujúce jednotlivé konštrukcie a vlastnosti schémy a porovnáme ju s pôvodnou schémou RSAProd.

V záverečnej šiestej kapitole zhrnieme dosiahnuté výsledky a uvedieme námety na ďalšiu prácu v tejto oblasti.

V kapitolách 2 a 3 sú zhrnuté už známe vedomosti o schémach CES. Obsahujú pojmy a konštrukcie potrebné na pochopenie obsahu kapitol 4 a 5. Ďalšie fakty o schémach CES možno nájsť v [1] a [3], z ktorých boli čerpané informácie použité pri písaní kapitol 2 a 3. V kapitolách 4, 5 a v časti 3.4 kapitoly 3 sú uvedené nové pojmy a konštrukcie, a takisto modifikácie už existujúcich schém a štruktúr.

Kapitola 2

Podpisové schémy

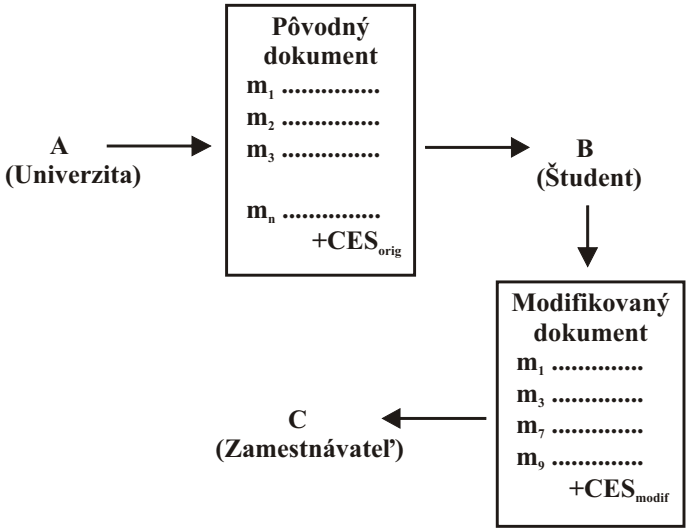
V tejto kapitole sa budeme zaoberať otázkou ako zväčšiť možnosti využitia digitálne podpísaných elektronických dokumentov. Konkrétne sa budeme venovať schémam, ktoré umožňujú vlastníkovi dokumentu odstraňovať časti pôvodného dokumentu pri zachovaní možnosti používateľa overiť autentickosť výsledného dokumentu.

Doteraz už bolo navrhnutých niekoľko schém, ktoré túto vlastnosť majú. Jeden prístup k problému prezentujú Johnson, Molnar, Song a Wagner v článku Homomorphic Signature Schemes [2]. Steinfeld, Bull a Zheng vo svojej práci Content Extraction Signatures [3] popisujú štyri nové schémy, dve založené na RSA (schémy RSAProd a MERSAProd) a ďalšie dve založené na všeobecných kryptografických primitívach (schémy CommitVector a HashTree). Podrobnejšie popíšeme iba schému RSAProd, ktorá bude tvoriť základ novej modifikovanej schémy uvedenej v kapitole 5.

2.1 Motivačný príklad

Uvedieme jednoduchý príklad toho, ako možno rozšíriť použitie elektronického dokumentu na rozmanité účely. Univerzita vydá študentovi po úspešnom ukončení jeho štúdia digitálne podpísaný formálny dokument – diplom. Študent pri hľadaní svojho budúceho zamestnávateľa musí k svojmu životopisu pripojiť aj dokumenty potvrdzujúce pravdivosť údajov v životopise. Napríklad na potvrdenie dosiahnutého vzdelania musí pripojiť overenú kópiu diplomu; v našom prípade kópiu elektronického dokumentu s platným digitálnym podpisom. Formálne dokumenty ako vysokoškolský diplom obsahujú okrem informácie o dosiahnutom vzdelaní aj osobné údaje o študentovi; napríklad dátum narodenia študenta. Aby uchádzač o zamestnanie predišiel diskriminácii na základe svojho veku, je len prirodzené, že nechce, aby

dátum jeho narodenia, ktorý sa nachádza na diplome bol pre zamestnávateľa viditeľný. Pri použití štandardného digitálneho podpisu musí byť diplom priložený k žiadosti v pôvodnej nezmenenej podobe, ináč by podpis nebolo možné overiť. Predpokladajme, že univerzita je ochotná akceptovať úspešnú verifikáciu diplomu s vynechaným dátumom narodenia, prípadne inými osobnými údajmi. Zároveň však vyžaduje, aby ostatné časti dokumentu boli vždy prítomné v každej overiteľnej kópii dokumentu.



Obr. 2.1: Príklad použitia CES

2.2 Jednoduché riešenie

Triválne riešenie problému spočíva v rozdelení diplomu na logické časti, napríklad každý nedeliteľný fakt bude zodpovedať jednej časti dokumentu. Následne univerzita všetky časti osobitne digitálne podpíše a študentovi odovzdá súhrn podpísaných fragmentov. Študent si potom vyberie, ktoré časti sa mu hodia pri komunikácii s treťou stranou a pošle na overenie iba zvolenú podmnožinu fragmentov pôvodného dokumentu.

Štandardná podpisová schéma sa skladá z troch algoritmov. Prvým je algoritmus `Keygen`, ktorý generuje verejný kľúč PK a súkromný kľúč SK. Druhý je algoritmus `Sign`, ktorý na základe dokumentu M a súkromného kľúča vyprodukuje príslušný podpis. Tretí algoritmus `Verify` na základe verejného kľúča overí pravosť podpisu daného dokumentu.

Jednoduchá schéma sa skladá zo štyroch algoritmov. Na jej zostrojenie možno použiť ľubovoľnú štandardnú podpisovú schému. Algoritmy `Sign` a `Verify` sa upravia tak, aby namiesto jedného dokumentu na vstupe mohli spracovať viac dokumentov (fragmenty pôvodného dokumentu). Pribudne ešte nový algoritmus `Extract`, ktorého úlohou je zostrojiť podpis pre nový (extrahovaný) dokument. V jednoduchej schéme tento algoritmus iba vyberie správnu podmnožinu z pôvodnej množiny podpisov.

Táto schéma má očividné slabiny. Jednak je to príliš veľká výpočtová zložitosť – pre každý fragment treba vypočítať osobitný digitálny podpis. Ďalej sú to zvýšené režijné náklady pri komunikácii – namiesto jedného podpisu treba prenášať medzi komunikujúcimi stranami toľko podpisov, koľko fragmentov obsahuje daná správa.

Vzniká prirodzená požiadavka vylepšiť túto schému. Zlepšiť možno:

- výpočtovú náročnosť
- režijné náklady pri komunikácii

Takisto si všimnime, že poradie fragmentov v dokumente, ktorý extrahuje vlastník z pôvodného dokumentu nemusí byť zachované. Rovnako nie sú fragmenty medzi sebou nijako previazané a ich extrahovanie je teda nezávislé od extrahovania ostatných fragmentov.

Predstavme si trochu prehnanú situáciu, že podpisovať sa bude každé slovo zvlášť – t.j. každé slovo bude samostatný fragment. Pri dostatočne dlhom pôvodnom dokumente budeme mať dostatočnú slovnú zásobu na to, aby sme vedeli z fragmentov poskladať text s úplne iným významom. Aj v situácii, keby sme podpisovali väčšie celky textu (napríklad vety) je určite nežiadúce, aby mohli v extrahovanom dokumente byť tieto vety v inom poradí ako boli v pôvodnom dokumente.

Teda okrem už vyššie spomenutých možných zlepšení je dobré, aby schéma mala aj tieto vlastnosti:

- schéma umožňuje podpisujúcej strane presne špecifikovať, ktoré podmnožiny množiny všetkých fragmentov je možné extrahovať
- schéma zachováva rovnakú bezpečnosť ako pri štandardnom digitálnom podpise (to znamená, že pri návrhu schémy použijeme štandardné kryptografické konštrukcie)

2.3 Terminológia

V tejto časti zavedieme základné definície, ktoré neskôr použijeme na definíciu schémy CES.

Popisované konštrukcie využívajú schému RSA, preto zavedieme nasledovné označenia. Symbolom d budeme označovať súkromný exponent, symbolom e verejný exponent. Kľúče budú dvojice $SK = (d, n)$ a $PK = (e, n)$, kde n je modulus danej inštancie schémy RSA.

Dokument, s ktorým budeme pracovať, označíme M. Budeme predpokladať, že je rozdelený na n fragmentov. Takisto budeme predpokladať, že má takú reprezentáciu, ktorá umožňuje číslovať jednotlivé fragmenty a teda kódovať ich poradie. Zápis $M = (m_1, m_2, \ldots, m_n)$ znamená, že dokument M sa skladá z n fragmentov, ktoré sú označené $m_1, m_2, \ldots, m_n$. Symbol X bude označovať množinu indexov tých fragmentov, ktoré budú extrahované do nového dokumentu M'. Dokument M' bude mať takisto n fragmentov ako dokument M; tie fragmenty, ktorých indexy sa nachádzali v množine X budeme nazývať plné, ostatné fragmenty budeme nazývať prázdne. Symbolom $M[i]$ budeme označovať i-ty fragment dokumentu M. A nakoniec $c(M)$ bude označovať množinu plných fragmentov dokumentu M; $c(M) = \{M[i]; i \in X\}$.

Napríklad, ak originálny dokument $M = (m_1, m_2, m_3, m_4)$ a množina $X = \{1, 3\}$, potom extrahovaný dokument $M' = (m_1, ?, m_3, ?)$, kde symbol ? označuje prázdny fragment. Množina $c(M') = \{1, 3\}$. Dokumenty $(m_1, m_3, ?, ?)$, $(m_3, ?, m_1, ?)$ aj $(m_1, ?, m_3, ?, ?)$ sú podľa definície rôzne od dokumentu M' a nie sú extrahovateľné z dokumentu M.

2.4 Funkčné požiadavky

Požiadavky, ktoré na schémy CES kladieme môžeme rozdeliť na 2 druhy. Sú to jednak požiadavky funkčné – teda tie, ktoré majú byť poskytnuté čestným používateľom schémy. Po druhé sú to požiadavky bezpečnostné, ktoré majú zabrániť nečestným používateľom schémy vykonávať neželané operácie. Funkčné požiadavky sú:

Možnosť extrahovania: Schéma CES musí umožňovať každému, kto má prístup k podpísanému pôvodnému dokumentu extrahovať nový dokument a nový verifikovateľný digitálny podpis k tomuto dokumentu bez interakcie s autorom pôvodného dokumentu a podpisu.

Túto požiadavku spĺňa aj jednoduché riešenie. Definujeme ďalšiu požiadavku, ktorá odlíši schémy CES od jednoduchej schémy.

Efektívnosť: Schéma CES musí byť efektívnejšia ako jednoduché riešenie buď v komunikačnej zložitosti (dĺžke pôvodného a/alebo extrahovaného podpisu) alebo vo výpočtovej zložitosti (pri podpisovaní alebo overovaní podpisu).

Ďalšia požiadavka, ktorá môže byť v niektorých aplikáciách užitočná, avšak nie je definíciou CES schémy explicitne vyžadovaná je požiadavka itero-

vaného extrahovania. Môžeme ju teda považovať za nepovinnú, aj keď určite nie za nezaujímavú.

Možnosť viacnásobného extrahovania: Schéma CES môže umožniť narábať s extrahovaným dokumentom a podpisom rovnako ako keby to bol originálny dokument, čím umožní viacnásobný proces extrahovania.

2.5 Bezpečnostné požiadavky

Najprv popíšme štandardnú požiadavku nesfalšovateľnosti podpisu. Táto požiadavka má zaručovať autenticitu a integritu podpísaného dokumentu. Keďže jedna z funkčných požiadaviek na schému CES hovorí, že držiteľ dokumentu má mať možnosť extrahovať z neho len určitú časť a vytvoriť k nej overiteľný digitálny podpis bez znalosti autorovho súkromného kľúča, musí sa definícia nesfalšovateľnosti prispôsobiť tejto funkčnej požiadavke.

Budeme však požadovať, aby autor pôvodného dokumentu (a podpisu) mal plnú kontrolu nad tým, ako môže vlastník tento dokumentu modifikovať. Autor môže žiadať, aby fragmenty boli medzi sebou viazané – teda, ak vlastník extrahuje nejaký fragment, bude musieť spolu s ním extrahovať aj ďalšie, aby sa predišlo zmene významu dokumentu vytrhnutím istých informácií z kontextu.

Autorove požiadavky budú zakódované v tzv. prístupovej štruktúre, ktorú budeme pre jednoduchosť označovať skratkou CEAS (z angl. Content Extraction Access Structure). Podrobne sa touto prístupovou štruktúrou budeme zaoberať v ďalšej kapitole. Zatiaľ nám postačí vedieť, že CEAS kóduje všetky možné podmnožiny fragmentov pôvodného dokumentu, ktoré je vlastník oprávnený z dokumentu extrahovať. S touto vedomosťou sme už schopní definovať požiadavku nesfalšovateľnosti pre schémy CES.

Nesfalšovateľnosť: Nie je možné zostrojiť takú dvojicu dokument—podpis (označme ju (M,s)), ktorá úšpešne vyhovie overovaciemu algoritmu `Verify` a zároveň bude spĺňať niektorú z nasledovných vlastností:

- M nie je platný extrahovaný dokument – teda nie je možné ho extrahovať zo žiadneho dokumentu doteraz podpísaného podpisovým algoritmom
- M je platný extrahovaný dokument (z pôvodného dokumentu D), avšak M nevyhovuje autorovým požiadavkam zakódovaným v prístupovej štruktúre CEAS pre dokument D

Daná definícia nesfalšovateľnosti zabraňuje pokusom o falšovanie v prípade, že extrahovaný dokument nie je platným extrahovaným dokumentom,

čo zahŕňa napríklad aj pokusy o útok preusporiadaním poradia fragmentov, takisto pokusom o útok opakovaním fragmentov a v neposlednom rade takisto aj pokusom o útok kompozíciou fragmentov, ktoré sa síce všetky vyskytli v niektorom z doteraz podpísaných dokumentov, avšak nie naraz v jednom dokumente.

Na rozdiel od klasických digitálnych podpisov, kde nesfalšovateľnosť je jediná bezpečnostná požiadavka, mnohé aplikácie schém CES môžu vyžadovať aj ďalšie bezpečnostné aspekty. Jednou z požiadaviek je napríklad utajenie. Vlastník dokumentu vymaže niektoré časti pôvodného dokumentu za účelom utajiť citlivé informácie. Nikde však nie je zaručené, že podpis, ktorý bude zostrojený k tomuto novému dokumentu nebude obsahovať nejakú (hoci aj nepoužiteľnú) časť pôvodnej citlivej informácie. Preto je potrebné formulovať ďalšiu bezpečnostnú požiadavku, ktorá bude riešiť tento problém.

Utajenie: Nech dokument $M_1 = (m_1, \ldots, M_{i1}, m_{i+1}, \ldots, m_n)$ a $M_2 = (m_1, \ldots, M_{i2}, m_{i+1}, \ldots, m_n)$. Množina X pre oba dokumenty nech je $X = \{1, 2, \ldots, i-1, i+1, \ldots, n\}$, nech D_1 a D_2 sú extrahované dokumenty prislúchajúce dokumentom M_1 a M_2 a množine X. Teda $D_1 = D_2$. Nech s_1 a s_2 sú extrahované podpisy prislúchajúce extrahovaným dokumentom D_1 a D_2. Potom podpisy s_1 a s_2 nie je možné od seba odlíšiť.

Táto požiadavka zjednodušene hovorí o tom, že o fragmentoch, ktoré neboli extrahované do nového dokumentu sa z extrahovaného podpisu nedá nič zistiť.

2.6 Schéma CES

Schéma CES: je štvorica algoritmov (`Keygen`, `Sign`, `Extract`, `Verify`), ktoré spĺňajú funkčné a bezpečnostné požiadavky uvedené v 2.4 a 2.5.

1. `Keygen` — algoritmus, ktorý vygeneruje dvojicu kľúčov (súkromný kľúč SK a verejný kľúč PK)

2. `Sign` — algoritmus dostane na vstupe súkromný kľúč SK, dokument $M = (m_1, m_2, \ldots, m_n)$ a prístupovú štruktúru $CEAS$; vráti podpis s_{FULL}

3. `Extract` — algoritmus na vstupe dostane dokument M, podpis s_{FULL}, množinu $X \subseteq \{1, 2, \ldots, n\}$ a verejný kľúč PK; vráti nový podpis s_{EXT} pre extrahovaný dokument

4. `Verify` — algoritmus dostane na vstupe extrahovaný dokument M', extrahovaný podpis s_{EXT} a verejný kľúč PK; na výstupe vracia odpoveď na otázku, či je daná dvojica dokument–podpis platná

Oproti štandardnej podpisovej schéme má táto nová schéma dve základné odlišnosti. Jednou je nový algoritmus `Extract`, ktorý sa v štandardnej podpisovej schéme vôbec nenachádza. Druhou odlišnosťou je použitie prístupovej štruktúry CEAS (vo všetkých algoritmoch okrem `Keygen`); štandardná podpisová schéma tento koncept nevyužíva.

2.7 Použitie schém CES

Aplikácie schém CES sa dajú rozdeliť do dvoch hlavných kategórií:

- **Ochrana súkromia (citlivých informácií):** Pôvodný dokument obsahuje citlivé informácie, ktoré si jeho vlastník nepraje zverejniť alebo poskytnúť tretím osobám.
- **Zefektívnenie prenosu dát:** Pôvodný dokument je veľmi dlhý, na overenie je však potrebná len jeho malá časť.

Uvedieme konkrétne príklady k obidvom situáciám. Na ilustráciu prvého prípadu nám veľmi dobre poslúži už spomenutá situácia, keď študent pri hľadaní zamestnania potrebuje preukázať dosiahnuté vzdelanie, avšak diplom vydaný univerzitou obsahuje aj citlivé údaje ako je dátum narodenia, ktoré chce študent pred zamestnávateľom zatajiť. Druhý prípad môžeme ilustrovať na nasledovnom príklade. Parlament podpíše zbierku zákonov jediným podpisom, pretože podpisovať znenie každého paragrafu zákona osobitne by bolo zdĺhavé a neprehľadné. Ak použije štandardný digitálny podpis, zbierka zákonov bude môcť byť citovaná (a overiteľná podpisom) iba celá. Ak použije na podpis schému CES (pričom paragrafy zákonov budú fragmenty dokumentu) môže každý citovať (a potvrdiť platným extrahovaným podpisom) ľubovoľnú množinu paragrafov.

2.8 Schéma RSAProd

V tejto sekcii uvedieme konkrétnu implementáciu schémy CES [3]. Schéma bude založená na RSA a oproti jednoduchému riešeniu bude efektívnejšia v komunikačnej zložitosti, presnejšie v dĺžke podpisu. Dĺžka pôvodného podpisu ostane nezmenená, avšak dĺžka extrahovaného podpisu bude oveľa kratšia – približne rovná dĺžke jediného podpisu pri klasickom použití RSA. Výpočtová zložitosť bude rovnaká ako pri jednoduchom riešení.

Myšlienka tejto schémy spočíva vo využití homomorfnej vlastnosti RSA, teda $h_1^d \cdot h_2^d \bmod N = (h_1 \cdot h_2)^d \bmod N$. Popíšme teraz jednotlivé algoritmy prislúchajúce tejto schéme. Algoritmus `Keygen` je úplne rovnaký ako štandardný

inicializačný algoritmus pre schému RSA. Podrobnejší popis ostatných algoritmov bude uvedený hneď za ich pseudokódom (symbol | označuje operáciu zreťazenia).

```
procedure Sign(M, CEAS, SK)
begin
   vypočítame alebo náhodne zvolíme CES-tag T
   h_i := H(CEAS|T|i|M[i]) pre i ∈ {1, ..., n}
   s_i := h_i^d mod N
   s_FULL := (CEAS|T|s_1|...|s_n)
end
```

Na začiatku algoritmus zvolí tzv. CES-tag, ktorý má byť vždy iný (doteraz nepoužitý). Sú dve základné možnosti na výpočet tagu. Prvou možnosťou je zvoliť CES-tag náhodne ako reťazec núl a jedničiek danej dĺžky. Pri tomto riešení si netreba pamätať doteraz použité tagy. Druhé riešenie je zaviesť pomocnú premennú – počítadlo, ktorého hodnota sa použije ako CES-tag a následne sa zvýši o 1 (alebo inú predvolenú hodnotu). Pri tomto riešení je zaručené, že sa nepoužije dvakrát ten istý tag.

V druhom kroku algoritmus ku každému fragmentu $M[i]$ pridá číslo fragmentu i, CES-tag T a prístupovú štruktúru $CEAS$. Na tento reťazec následne aplikuje hašovaciu funkciu H. Týmto sa zaistí previazanie daného fragmentu s jeho poradovým číslom (zabraňujúc tak útoku opakovaním fragmentov alebo zmenou ich poradia), takisto sa fragment previaže s prístupovou štruktúrou $CEAS$ (zabraňujúc nežiadanému extrahovaniu fragmentov) a nakoniec sa fragment previaže aj CES-tagom T (zabraňujúc miešaniu fragmentov z rôznych dokumentov).

V treťom kroku sa vypočítajú zložky výsledného podpisu. V tomto kroku (a nikde inde) sa používa súkromná informácia autora podpisu – konkrétne súkromný exponent d.

Výsledný podpis, ktorý sa vracia v štvrtom kroku algoritmu je zreťazením jednotlivých zložiek podpisu vypočítaných v treťom kroku, CES-tagu T a prístupovej štruktúry $CEAS$.

```
procedure Extract(M, s_FULL, X, PK)
begin
   rozdeľme s_FULL na zložky: s_FULL = (CEAS|T|s_1|...|s_n)
   s_f := ∏_{i∈X} s_i mod N
   s_EXT := T|CEAS|s_f
end
```

Algoritmus dostane na vstupe pôvodný podpis s_{FULL}, ktorý rozdelí na jednotlivé časti prislúchajúce fragmentom dokumentu. Následne vypočíta súčin tých čiastkových podpisov, ktorých indexy sa nachádzajú v množine X. Týmto sa z množiny čiastkových podpisov stane jediný čiastkový podpis prislúchajúci množine X. Výsledný extrahovaný podpis je zreťazením tohto čiastkového podpisu, CES-tagu T a prístupovej štruktúry $CEAS$. Z toho teda vyplýva, že extrahovaný podpis má približne dĺžku jediného štandardného podpisu – presnejšie je to dĺžka jedného podpisu plus dĺžka $CEAS$ plus dĺžka CES-tagu.

Vďaka operácii `Extract` na čiastkových podpisoch sa extrahovaný podpis nedá použiť ako nový originálny podpis. Teda táto schéma nespĺňa nepovinnú požiadavku na CES schémy o možnosti viacnásobného extrahovania z dokumentu.

```
procedure Verify(M', s_EXT, PK)
begin
   X := c(M')
   rozdeľme s_EXT na zložky: s_EXT = T|CEAS|s_f
   for i ∈ X: h_i := H(CEAS|T|i|M'[i])
   if (s_f^e = ∏_{i∈X} h_i mod N) and (X ∈ CEAS)
   . verifikácia prebehla úspešne
   else
   . verifikácia zlyhala
end
```

Na začiatku algoritmus zistí, ktoré fragmenty sú plné (nezmenené pri extrahovaní) a ich indexy uloží do množiny X. Extrahovaný podpis s_{EXT}, ktorý dostal na vstupe rozdelí na CES-tag, prístupovú štruktúru a čiastkový podpis. Následne – podobne ako algoritmus `Extract` – vypočíta hodnoty h_i, ktoré však tentokrát zodpovedajú fragmentom extrahovaného dokumentu. Test, ktorý vykoná v ďalšom kroku využíva homomorfnú vlastnosť RSA a je ťažiskovým bodom celej tejto schémy. Ak test na rovnosť dopadne pozitívne, stačí už iba skontrolovať, či boli dodržané všetky požiadavky autora dokumentu zakódované v prístupovej štruktúre CEAS. Ak aspoň jeden z týchto dvoch testov zlyhá, potom zlyhá celá verifikácia podpisu.

Kapitola 3

Prístupové štruktúry

V kapitole 2 sme uviedli definíciu schémy CES, príklady jej použitia a jednu možnú implementáciu schémy. Odvolávali sme sa na existenciu tzv. prístupovej štruktúry CEAS. V tejto kapitole sa budeme CEAS-om podrobne zaoberať.

Rozoberme dôvody, ktoré nás viedli k tomu, aby sme sa zamýšľali nad rôznymi prístupmi k tomuto problému. Prvým dôvodom je otázka priestorovej náročnosti. V časti 2.8 sme uviedli, že extrahovaný podpis má dĺžku približne ako jeden klasický podpis (za predpokladu, že dĺžka CES-tagu a dĺžka kódovania CEAS-u sú zanedbateľné alebo nanajvýš porovnateľné s dĺžkou čiastočného podpisu prislúchajúcemu množine X). Keďže dĺžka CES-tagu je konštantná, stačí sa zaoberať dĺžkou zápisu CEAS-u. Tá závisí od požiadaviek, ktoré budeme na funkcionalitu CEAS-u klásť. Od dĺžky pôvodného dokumentu (presnejšie od počtu fragmentov n tohto dokumentu) môže dĺžka zápisu CEAS-u závisieť lineárne, kvadraticky, dokonca aj exponenciálne.

Ďalší dôvod je poskytnutie čo najväčších možností autorovi dokumentu pri definovaní prístupovej štruktúry. Čím viac možností autorovi poskytneme, tým bude prístupová štruktúra zložitejšia. Úlohou bude nájsť vhodný kompromis medzi dĺžkou CEAS-u a autorovými možnosťami.

Nakoniec, je vhodné zamyslieť sa aj nad konkrétnym využitím schémy CES. Ako sme uviedli v kapitole 2, existujú dve hlavné kategórie využitia. Pri využití schémy na ochranu citlivých informácií, vlastník dokumentu väčšinou extrahuje z pôvodného dokumentu veľkú väčšinu textu a vynechá len malé množstvo informácie. Naopak pri využití schémy na zefektívnenie prenosu dát sa z pôvodného dokumentu extrahuje iba malá časť.

3.1 Typy štruktúr

Ak chceme umožniť autorovi definovať ľubovoľnú množinu povolených kombinácií fragmentov, musí byť prístupová štruktúra schopná zakódovať celú potenčnú množinu množiny fragmentov. Teda maximálna dĺžka CEAS-u bude 2^n. Malé vylepšenie spočíva v tom, že keď bude množina povolených kombinácií väčšia ako množina nežiadúcich kombinácií, môžeme zakódovať tú druhú množinu – keďže sú navzájom komplementárne. Takto dosiahneme maximálnu dĺžku CEAS-u 2^{n-1}, čo je však stále exponenciálna zložitosť. Toto je samozrejme neprijateľné; o využití tejto možnosti kódovania sa dá uvažovať iba v prípade, že jedna z týchto množín je veľmi malá (maximálne niekoľko desiatok prvkov).

Ďalší parameter, ktorý sme spomenuli bolo konkrétne použitie danej schémy. V prípade, že je schéma používaná na ochranu citlivých informácií môže prístupová štruktúra pozostávať z čísel fragmentov zodpovedajúcich týmto informáciám. Pri extrahovaní dokumentu fragmenty uvedené v tejto množine môžu byť vynechané, ostatné musia ostať nezmenené. Dĺžka prístupovej štruktúry bude nanajvýš rovná počtu fragmentov dokumentu. Jedná sa teda o lineárnu zložitosť. Ak využijeme trik z predchádzajúceho riešenia, môžeme maximálnu dĺžku CEAS-u ohraničiť číslom $\lfloor n/2 \rfloor$.

Práve spomenuté riešenie však nerieši otázku vytrhnutia informácie z kontextu, keďže medzi jednotlivými fragmentami nie sú žiadne väzby. Naopak exponenciálne riešenie tento problém rieši až príliš zložito. Ak upustíme od požiadavky, aby bolo možné definovať úplne ľubovoľnú podmnožinu kombinácií (potenčnej množiny), môžeme dostať oveľa prijateľnejšiu zložitosť. Okrem pojmu fragment zavedieme ďalší pojem – väzba. Tento pojem bude reprezentovať vzťah medzi dvoma fragmentami. Riešenia tohto typu majú kvadratickú zložitosť a popíšeme ich v časti 3.3.

3.2 Jednorozmerná prístupová štruktúra

Vráťme sa znova ku príkladu s univerzitou, ktorá vydáva študentovi digitálne podpísaný diplom. Univerzita chce umožniť študentovi zatajiť niektoré časti dokumentu – dátum narodenia, rodné číslo, prípadne iné citlivé informácie – avšak zvyšok dokumentu musí ostať v pôvodnom znení. Pritom sa neberie ohľad na kontext; medzi fragmentami nie sú žiadne väzby. Štruktúra fyzickej reprezentácie CEAS-u môže byť preto veľmi jednoduchá.

Jednorozmerná prístupová štruktúra: je reprezentovaná bitovým vektorom $v = (v_1, \ldots, v_n) \in \{0,1\}^n$. Ak $v_i = 0$ znamená to, že i-ty fragment je povinný (musí ostať nezmenený v každom extrahovanom dokumente). Ak

$v_i = 1$, tak i-ty fragment dokumentu je nepovinný (môže byť v extrahovanom dokumente nahradený prázdnym fragmentom).

Táto reprezentácia je na jednej strane veľmi úsporná, avšak na strane druhej nie je použiteľná pri komplikovanejších požiadavkach autora.

3.3 Viacrozmerná prístupová štruktúra

Reportér sa po interview so známou osobnosťou hodlá poslať tento rozhovor do redakcie. Známa osobnosť však požaduje, aby jej slová nemohli byť vytrhnuté z kontextu a musia byť presne citované. Táto situácia sa dá vyriešiť použitím digitálneho podpisu, ktorý si budú čitatelia elektronického článku môcť overiť. Z rôznych dôvodov (neúnosná dĺžka interview, nevhodné otázky, nevhodné členenie rozhovoru) chce redakcia urobiť určité zmeny pri kompilovaní výsledného článku (napríklad vynechať niektoré otázky a odpovede, prípadne zmeniť ich poradie). Zároveň však musí rešpektovať požiadavky známej osobnosti, ktorá chce udržať otázky a odpovede v kontexte.

V tejto situácii je výhodné použiť na podpísanie interview schému CES. Pôvodný dokument rozdelíme na fragmenty zodpovedajúce jednotlivým otázkam a odpovediam. Okrem fragmentov definujeme aj tzv. väzby medzi fragmentami, ktoré znemožnia neželané manipulovanie.

Okrem väzby medzi danou otázkou a zodpovedajúcou odpoveďou môžeme uvažovať širší kontext. Môže byť žiaduce zoskupovať jednotlivé dvojice otázka–odpoveď do väčších skupín. V tejto skupine môže byť niektorá otázka ťažisková, ostatné ju buď musia alebo môžu doprevádzať.

Je potrebné vedieť špecifikovať, ktoré fragmenty:

- môžu byť extrahované samostatne
- môžu byť extrahované iba keď sú doprevádzané iným fragmentom alebo fragmentami
- môžu byť voliteľne extrahované doprevádzajúc iný fragment alebo fragmenty
- nemôžu byť extrahované – teda môžu byť poskytnuté iba spolu s celým dokumentom

Fragmenty rozdelíme do dvoch skupín: na primárne a sekundárne. Primárny fragment môže byť extrahovaný do nového dokumentu priamo. Ak fragment je nie primárny, tak je automaticky sekundárny. Sekundárne fragmenty môžu byť do nového dokumentu extrahované iba nepriamo, cez väzbu s niektorým primárnym fragmentom.

Medzi fragmentami existujú väzby dvoch druhov: povinné a nepovinné. Jeden fragment môže mať žiadnu, jednu alebo aj viacero väzieb s ostatnými fragmentami. Medzi dvoma fragmentami môže byť nanajvýš jedna z týchto väzieb, nemôžu byť obe súčasne. Väzba je asymetrická a tranzitívna.

Popíšme si teraz jednotlivé situácie, ktoré môžu pri zoskupovaní nastať:

- fragment je primárny a nemá žiadne väzby — tento fragment môže byť priamo extrahovaný do nového dokumentu a neovplyvní ostatné fragmenty
- fragment je primárny a má povinnú väzbu — ak je tento fragment extrahovaný musia byť spolu s ním extrahované všetky fragmenty zviazané povinnou väzbou
- fragment je primárny a má nepovinnú väzbu — ak je tento fragment extrahovaný môžu ho doprevádzať fragmenty s ním zviazané nepovinnou väzbou
- fragment je sekundárny a nemá žiadnu väzbu — tento fragment nie je možné extrahovať, teda môže byť poskytnutý iba spolu s celým pôvodným dokumentom
- fragment je sekundárny a má povinnú väzbu — ak je fragment na druhom konci väzby primárny a bol extrahovaný, musí byť extrahovaný aj tento fragment
- fragment je sekundárny a má nepovinnú väzbu — ak je fragment na druhom konci väzby primárny a bol extrahovaný, môže byť extrahovaný aj tento fragment

3.4 Grafová reprezentácia

Existuje viacero možností fyzickej reprezentácie prístupovej štruktúry. Niekoľko konkrétnych implementácií uvádzajú Bull, Squire a Zheng v článku A Hierarchical Extraction Policy for Content Extraction Signatures [1]. V stručnosti popíšeme maticovú implementáciu pre viacrozmernú prístupovú štruktúru. Z nej potom odvodíme grafovú reprezentáciu prístupovej štruktúry.

Viacrozmerná prístupová štruktúra: je reprezentovaná bitovou maticou M typu $n \times n$ (n je počet fragmentov dokumentu). Ak $M[i,i] = 1$, i-ty fragment primárny, v opačnom prípade ($M[i,i] = 0$) je sekundárny. Ak $M[i,j] = 1 \wedge M[i,i] = 1$ $(i \neq j)$ medzi fragmentami i a j existuje povinná väzba. Ak $M[i,j] = 1 \wedge M[i,i] = 0$ $(i \neq j)$ medzi fragmentami i a j existuje

nepovinná väzba. Ak $M[i,j] = 0$ $(i \neq j)$ medzi fragmentami i a j neexistuje väzba.

Teda prvky matice na hlavnej diagonále určujú, či je fragment primárny alebo sekundárny. Ostatné prvky určujú, či medzi fragmentami existujú väzby a zároveň spolu s diagonálnymi prvkami určujú druh väzby.

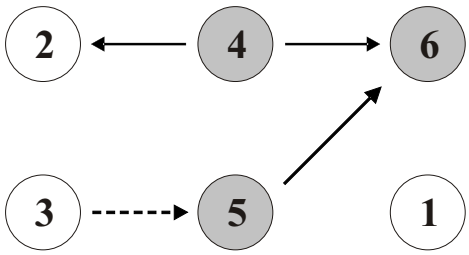
Číslo fragmentu	CEAS
1	0 0 0 0 0 0
2	0 0 0 0 0 0
3	0 0 0 0 1 0
4	0 1 0 1 0 1
5	0 0 0 0 1 1
6	0 0 0 0 0 1

Obr. 3.1: Príklad implementácie CEAS-u maticou

Keďže väzby medzi fragmentami sú exkluzívne (teda väzba medzi dvoma fragmentami je buď povinná alebo nepovinná) a asymetrické, matica predstavujúca fyzickú reprezentáciu prístupovej štruktúry môže byť chápaná ako matica susedností orientovaného grafu.

Grafová reprezentácia prístupovej štruktúry: je ohodnotený orientovaný graf $G = (V, E, f)$. Množina vrcholov $V = \{v_1, \ldots v_n\}$ zodpovedá množine fragmentov dokumentu. Množina usporiadaných dvojíc vrcholov (orientovaných hrán) $E = \{e_1, \ldots e_m\}$ zodpovedá väzbám medzi fragmentami. Ak existuje hrana (slučka) $e_j = (v_i, v_i)$, tak fragment zodpovedajúci vrcholu v_i je primárny, v opačnom prípade je sekundárny. Ak existuje hrana $e_k = (v_i, v_j)$, potom existuje väzba viažuca j-ty fragment k i-temu fragmentu. Hodnotová funkcia $f : E \rightarrow \{0, 1\}$ určuje druh väzby. Ak $f(e_j) = 1$, hrana e_j zodpovedá povinnej väzbe. Ak $f(e_j) = 0$ hrana e_j zodpovedá nepovinnej väzbe.

Namiesto použitia hodnotovej funkcie môžeme povinné väzby kresliť plnou čiarou a nepovinné väzby prerušovanou čiarou. Ak navyše rozlíšime (napr. farebne) primárne fragmenty od sekundárnych, dostávame užívateľsky príjemný a dostatočne prehľadný nástroj na definovanie prístupovej štruktúry. Pre jednoduchosť a zrozumiteľnosť budeme odteraz používať takúto reprezentáciu. Grafovú reprezentáciu matice z predošlého príkladu ilustruje nasledovný obrázok.



Obr. 3.2: Grafová reprezentácia CEAS-u

Kapitola 4

Modifikácie prístupových štruktúr

V predchádzajúcej kapitole sme uviedli niekoľko reprezentácií prístupovej štruktúry. Jednoduché riešenie uvedené v časti 3.2 neumožňuje väzby medzi fragmentami. Riešenia, ktoré sa zaoberajú aj väzbami majú kvadratickú zložitosť.

V tejto kapitole vytvoríme novú reprezentáciu prístupovej štruktúry. V časti 4.1 uvedieme riešenie, ktoré pracuje aj s väzbami medzi fragmentami s lineárnou priestorovou zložitosťou. V časti 4.2 navrhneme implementáciu tejto štruktúry. Výhody a nevýhody zhrnieme v časti 4.3 a nakoniec v časti 4.4 uvedieme praktické príklady použitia tejto štruktúry.

4.1 Stromová prístupová štruktúra

Ako sme už spomenuli v úvode, chceme vytvoriť prístupovú štruktúru, ktorá vie pracovať s väzbami medzi fragmentami a zároveň jej zložitosť je v najhoršom prípade lineárna. Toto možno dosiahnuť obmedzením výskytu väzieb. Popíšme ako bude nová prístupová štruktúra vyzerať. Pre zjednodušenie využijeme grafovú reprezentáciu štruktúry tak, ako sme ju zaviedli v predchádzajúcej kapitole.

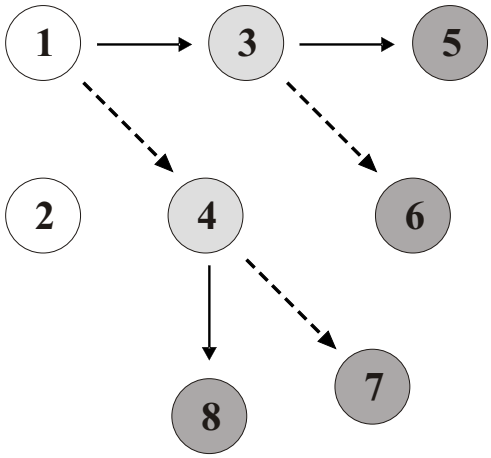
Pôvodný dokument M je rozdelený na n fragmentov, ktorým zodpovedá n vrcholov grafu. V predchádzajúcich reprezentáciách sme fragmenty rozdelili na primárne a sekundárne. V tomto prípade zavedieme jemnejšie delenie fragmentov. Podľa vzdialenosti vrcholu od určeného význačného vrcholu budeme vrcholy (fragmenty) deliť na vrcholy prvej úrovne, druhej úrovne, tretej úrovne atď. Fragmenty prvej úrovne budú tie, ktoré možno do nového dokumentu extrahovať priamo. Fragmenty druhej úrovne možno extrahovať

iba cez väzbu s fragmentom prvej úrovne. Podobne fragmenty tretej úrovne možno extrahovať iba cez väzbu s fragmentom druhej úrovne.

Podobne ako v predchádzajúcich reprezentáciách, väzba je asymetrická relácia medzi fragmentami. Premietnuté do teórie grafov, väzbu bude znázorňovať orientovaná hrana medzi dvoma vrcholmi (resp. z jedného vrcholu do druhého). Ak existuje hrana z vrcholu A do vrcholu B, potom nemôže existovať opačná hrana (z vrcholu B do vrcholu A). Väzby budú dvoch druhov: povinné a nepovinné. Povinné väzby budú graficky znázornené plnou hranou, nepovinné prerušovanou hranou. Väzby sú exkluzívne, to znamená, že medzi dvoma fragmentami nemôže byť zároveň povinná aj nepovinná väzba.

Obmedzenie výskytu väzieb spočíva v tom, že do každého vrcholu môže vchádzať maximálne jedna orientovaná hrana. Do vrcholu prvej úrovne nebude vchádzať žiadna hrana, do ostatných bude vchádzať práve jedna hrana. Vychádzať z vrcholu môže ľubovoľný počet hrán. Hrany vždy smerujú z vrcholu i-tej úrovne do vrcholu $(i + 1)$-vej úrovne.

Ak si uvedomíme, čo tieto obmedzenia znamenajú zistíme, že výsledný graf bude mať stromovú štruktúru. Graf bude pozostávať z jedného alebo viacerých komponentov súvislosti — stromov (ak nahradíme orientované hrany neorientovanými). Konkrétny príklad prístupovej štruktúry tohto typu je uvedený na obrázku 4.1.



Obr. 4.1: Grafová reprezentácia stromovej prístupovej štruktúry

Popíšme podrobne jednotlivé fragmenty na obrázku, väzby medzi nimi a vyplývajúce súvislosti.

- Fragment 1 — je fragment prvej úrovne. Je teda možné ho priamo extrahovať do nového dokumentu. Spolu s ním musí byť extrahovaný

aj fragment 3, pretože je s ním zviazaný povinnou väzbou. Fragment 4 viazaný nepovinnou väzbou môže a nemusí byť extrahovaný.

- Fragment 2 — je fragment prvej úrovne. Keďže nemá žiadne väzby môže byť extrahovaný nezávisle od ostatných fragmentov.
- Fragment 3 — je fragment druhej úrovne. Môže byť extrahovaný iba sprevádzajúc fragment 1. Ak však je extrahovaný musí ho sprevádzať fragment tretej úrovne s číslom 5 (kvôli povinnej väzbe). Fragment 6 ho môže a nemusí doprevádzať.
- Fragment 4 — je fragment druhej úrovne. Môže byť extrahovaný sprevádzajúc fragment 1. Ak je však extrahovaný musí ho doprevádzať fragment 8. Fragment 7 ho môže a nemusí doprevádzať.
- Fragment 5 — je fragment tretej úrovne. Môže byť extrahovaný iba sprevádzajúc fragment druhej úrovne, s ktorým je zviazaný (v tomto prípade povinnou väzbou s fragmentom 3).
- Fragment 6 — je fragment tretej úrovne. Môže a nemusí doprevádzať fragment 3 za predpokladu, že je extrahovaný.
- Fragment 7 — je fragment tretej úrovne. Môže doprevádzať fragment 4, opäť len za predpokladu, že bol extrahovaný.
- Fragment 8 — je fragment tretej úrovne. Musí byť extrahovaný spolu s fragmentom 4 (ak bol extrahovaný).

4.2 Implementácia

Stromová prístupová štruktúra: je reprezentovaná jednorozmerným poľom A dĺžky n (n je počet fragmentov). Jednotlivé položky poľa sú celé čísla z intervalu $-n$ až n. Ak $A[i] = k$ ($k \neq 0$), potom medzi fragmentami s číslami i a $|k|$ je väzba (v smere z $|k|$ do i). Znamienko čísla k bude určovať druh väzby: kladné čísla budú reprezentovať povinné väzby, záporné čísla nepovinné väzby.

Obmedzenie výskytu väzieb v stromovej štruktúre (konkrétne vlastnosť, že do vrcholu vstupuje maximálne jedna väzba) zaručuje, že rôzne väzby budú prislúchať rôznym prvkom poľa. Takto pri dokumente s n fragmentami a prístupovou štruktúrou s j komponentami zaplníme $n - j$ prvkov poľa. Zvyšných j prvkov poľa bude zodpovedať fragmentom prvej úrovne, do ktorých nesmeruje žiadna väzba. Tieto prvky budú definitoricky rovné nule.

Takto môžeme zároveň priamo rozlíšiť fragmenty prvej úrovne od ostatných fragmentov.

Dosiahli sme lineárnu veľkosť fyzickej reprezentácie prístupovej štruktúry s možnosťou využiť väzby medzi fragmentami. Zmenšenie výslednej veľkosti extrahovaného podpisu je možné dosiahnuť aj ináč. Stačí obmedziť počet väzieb v štruktúre na $\sqrt{n}$. Tým síce dosiahneme rovnakú zložitosť, avšak táto trieda štruktúr neobsahuje veľa prakticky použiteľných štruktúr. Navyše v kapitole 5 ukážeme, že stromovú prístupovú štruktúru je možné z extrahovaného podpisu dokonca úplne odstrániť.

Na ilustráciu tejto reprezentácie uveďme príklad zodpovedajúci prístupovej štruktúre uvedenej v predchádzajúcom príklade grafovou reprezentáciou.

Číslo fragmentu / Index poľa	Hodnota v poli
1	0
2	0
3	1
4	-1
5	3
6	-3
7	-4
8	4

Obr. 4.2: Implementácia CEAS-u pomocou poľa

4.3 Výhody a nevýhody

Prvou výhodou stromovej štruktúry je kombinácia lineárnej zložitosti spolu s možnosťou autora definovať väzby medzi fragmentami.

Druhou výhodou tejto štruktúry je, že môže byť zakomponovaná do pôvodného (neextrahovaného) podpisu. Touto operáciou sa síce zväčší veľkosť originálneho podpisu, avšak výrazne sa zmenší veľkosť extrahovaného podpisu, v ktorom už prístupová štruktúra nebude vôbec vystupovať. Keď si uvedomíme, že komunikácia medzi autorom dokumentu a vlastníkom dokumentu prebehne iba raz, zatiaľ čo komunikácia medzi vlastníkom a treťou stranou (overujúcou extrahovaný podpis) môže prebehnúť veľakrát, tento kompromis je určite prínosný. Podrobne sa touto operáciou budeme zaoberať pri modifikácii schémy RSAProd v kapitole 5.

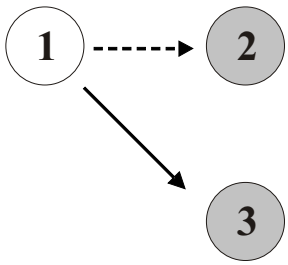
Treťou výhodou (oproti doterajším riešeniam) je možnosť zoskupovania fragmentov vo viacerých úrovniach. Riešenia z predchádzajúcej kapitoly po-

čítali iba s fragmentami primárnymi (prvej úrovne) a sekundárnymi (druhej úrovne). Stromová štruktúra umožňuje definovať väzby do ľubovoľnej hĺbky, čím lepšie uspokojuje potreby pri definovaní väzieb v štruktúrovaných dokumentoch.

Na druhej strane treba spomenúť aj nevýhodu tejto štruktúry — obmedzenie výskytu väzieb medzi fragmentami. Táto reštriktívna vlastnosť zmenšuje množinu potenciálnych prístupových štruktúr; stále však umožňuje zostrojiť veľa prakticky použiteľných štruktúr. Na ilustráciu možností tohto riešenia uvedieme grafové reprezentácie prístupových štruktúr pre všetky tri doteraz spomenuté príklady reálneho použitia štruktúry.

4.4 Príklady

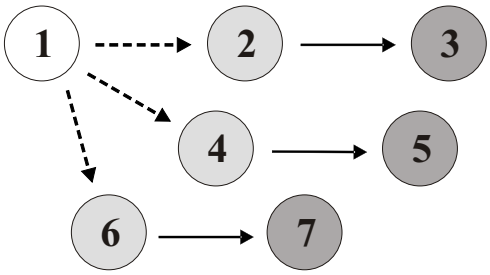
Prvý príklad (obr. 4.3) je veľmi jednoduchý. Jedinou požiadavkou autora dokumentu (univerzity) je, aby študent mohol z dokumentu (diplomu) odstrániť citlivú informáciu (dátum narodenia). Dokument teda rozdelí na 3 fragmenty. Prvý fragment je časť dokumentu od začiatku až po dátum narodenia, druhý fragment je samotný dátum narodenia a tretí fragment je zvyšná časť dokumentu. Jediným fragmentom prvej úrovne je v tomto prípade počiatočná časť dokumentu. Ak je extrahovaná, musí ju doprevádzať fragment 3. Teda minimálny možný nový dokument, ktorý je možné extrahovať, je pôvodný dokument bez dátumu narodenia. Ak si vlastník dokumentu želá extrahovať aj túto citlivú informáciu (fragment 2), je mu to umožnené cez nepovinnú väzbu tohto fragmentu s fragmentom s číslom 1.



Obr. 4.3: Príklad: Univerzita–študent–zamestnávateľ

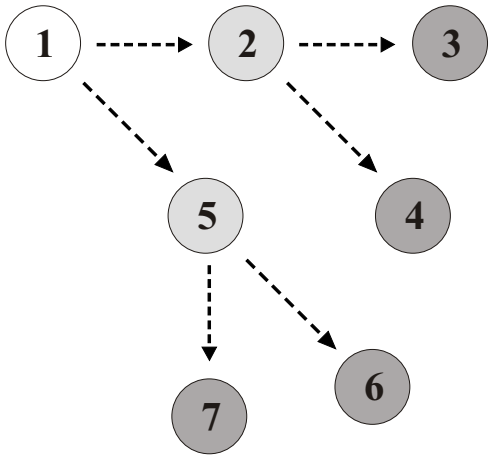
Druhý príklad je znázornený na obrázku 4.4. Dokument s obsahom interview môže mať nasledovnú štruktúru: začína hlavičkou dokumentu (meno reportéra, meno hosťa, dátum, ...) a pokračuje dvojicami otázka a odpoveď. V našom príklade hlavička dokumentu je fragment prvej úrovne s číslom 1, otázky sú fragmenty druhej úrovne s číslami 2, 4, 6 a odpovede sú fragmenty

tretej úrovne s číslami 3, 5 a 7. Je žiaduce, aby sa hlavička dokumentu vyskytovala v každom extrahovanom dokumente. To sme zaručili tým, že každý ďalší fragment sa dá extrahovať iba cez väzbu s fragmentom 1 alebo cez väzbu s iným fragmentom, ktorý má väzbu s fragmentom 1. Po extrahovaní hlavičky si môžeme zvoliť ľubovoľnú podmnožinu množiny otázok, ktoré chceme extrahovať, keďže všetky fragmenty zodpovedajúce otázkam sú na hlavičku viazané nepovinnou väzbou. Po extrahovaní otázok musíme extrahovať aj k nim prislúchajúce odpovede, ktoré sú na otázky naviazané povinnou väzbou.



Obr. 4.4: Príklad: Interview

Posledný príklad je znázornený na obrázku 4.5. Štruktúra zbierky zákonov môže byť napríklad takáto: dokument začína hlavičkou (rok vydania zbierky, počet zákonov, ...) a pokračuje zneniami jednotlivých zákonov. Zákony sú ešte vnútorne štruktúrované a takisto začínajú hlavičkou zákona a pokračujú zneniami jednotlivých paragrafov. Našim úmyslom je vytvoriť takú prístupovú štruktúru, aby sa dal citovať (extrahovať) každý paragraf zvlášť, avšak musí byť doprevádzaný hlavičkou zákona, do ktorého patrí a takisto hlavičkou zbierky, v ktorej bol zákon uverejnený. Tento úmysel presne vystihuje prístupová štruktúra uvedená na obrázku. Ak chceme extrahovať nejaký paragraf, musíme začať od fragmentu prvej úrovne (hlavičkou zbierky). Následne si vyberieme zákon, v ktorom sa nachádza daný paragraf a cez nepovinnú väzbu s fragmentom 1 extrahujeme hlavičku tohto zákona (fragment druhej úrovne). Nakoniec cez nepovinnú väzbu môžeme extrahovať aj požadovaný fragment tretej úrovne (daný paragraf).



Obr. 4.5: Príklad: Zbierka zákonov

Kapitola 5

Modifikácia RSAProd

Pôvodná schéma RSAProd prenášala prístupovú štruktúru medzi jednotlivými účastníkmi komunikácie osobitne. Na ochranu pred neoprávnenou zmenou prístupovej štruktúry vlastníkom dokumentu schéma požadovala CEAS ako jednu zložku vstupu pre hašovaciu funkciu H. Tým sa informácia o prístupovej štruktúre zakódovala priamo do čiastkových podpisov jednotlivých fragmentov. Vlastník dokumentu síce mohol overovateľovi podpisu poslať falošný CEAS, tým mu však znemožnil úspešné overenie podpisu, keďže overovateľ ho používa na spätnú rekonštrukciu výstupu hašovacej funkcie.

Modifikovaná schéma bude so stromovou prístupovou štruktúrou narábať iným spôsobom. Algoritmus `Sign` už nebude medzi vstupy hašovacej funkcie zahŕňať CEAS, a teda ani algoritmus `Verify` nebude musieť poznať prístupovú štruktúru na to, aby mohol úspešne rekonštruovať výstup hašovacej funkcie. Prístupová štruktúra teda nebude súčasťou extrahovaného podpisu.

Hlavnou myšlienkou tejto modifikácie je vytvoriť také čiastkové podpisy, ktoré bude môcť vlastník dokumentu kombinovať iba určitým spôsobom. Využijeme pritom nasledovnú rovnosť:

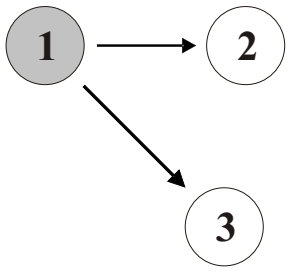
$$h_1^{d-x}(h_2^d h_1^x) = h_1^d h_2^d$$

Predtým, než uvedieme presné znenie modifikovaných algoritmov, ilustrujeme princíp fungovania tejto schémy na dvoch jednoduchých príkladoch. V časti 5.1 popíšeme zakomponovanie povinných väzieb do extrahovaného podpisu, v časti 5.2 zakomponovanie nepovinných väzieb. Zovšeobecnenie tohto postupu a konkrétnu implementáciu schémy uvedieme v časti 5.3.

5.1 Povinné väzby

Na obrázku 5.1 je fragment prvej úrovne povinnou väzbou previazaný s dvoma fragmentami druhej úrovne. Predpokladajme, že čiastkové podpisy prislúchajúce týmto fragmentom v pôvodnej schéme sú

$$s_1 = h_1^d \bmod N$$
$$s_2 = h_2^d \bmod N$$
$$s_3 = h_3^d \bmod N$$



Obr. 5.1: Príklad 1 - povinné väzby

Obohaťme fragmenty, v ktorých končia povinné väzby náhodným číslom. V tomto prípade sú to fragmenty s číslami 2 a 3. Zodpovedajúce náhodné čísla označme x a y. Skonštruujme nové čiastkové podpisy, ktoré budú mať nasledovný tvar

$$\begin{aligned} s'_1 &= h_1^{d-x-y} \bmod N \\ s'_2 &= h_2^d h_1^x \bmod N \\ s'_3 &= h_3^d h_1^y \bmod N \end{aligned}$$

Rozoberme dve situácie. Ak dodržíme prístupovú štruktúru a extrahujeme do nového dokumentu všetky 3 fragmenty, výsledný extrahovaný podpis bude mať tvar

$$s'_{EXT} = h_1^d h_2^d h_3^d \bmod N$$

To je presne rovnaký podpis ako by sme dostali pri použití klasickej schémy RSAProd. Ak nedodržíme prístupovú štruktúru výsledný extrahovaný podpis nesmie byť overiteľný. V nasledujúcej tabuľke je prehľad možných situácií.

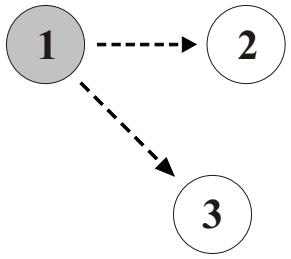
Prvý	Druhý	Tretí	Súčin	Overiteľný?
s'_1			h_1^{d-x-y}	Nie
	s'_2		$h_1^x h_2^d$	Nie
		s'_3	$h_1^y h_3^d$	Nie
s'_1	s'_2		$h_1^{d-y} h_2^d$	Nie
s'_1		s'_3	$h_1^{d-x} h_3^d$	Nie
	s'_2	s'_3	$h_1^{x+y} h_2^d h_3^d$	Nie
s'_1	s'_2	s'_3	$h_1^d h_2^d h_3^d$	Áno

Obr. 5.2: Prehľad možných kombinácií čiastkových podpisov

5.2 Nepovinné väzby

Na obrázku 5.3 je fragment prvej úrovne previazaný s dvoma fragmentami druhej úrovne nepovinnou väzbou. Predpokladajme, že čiastkové podpisy prislúchajúce týmto fragmentom v pôvodnej schéme sú

$$s_1 = h_1^d \bmod N$$
$$s_2 = h_2^d \bmod N$$
$$s_3 = h_3^d \bmod N$$



Obr. 5.3: Príklad 2 - nepovinné väzby

Obohaťme fragmenty, v ktorých začínajú nepovinné väzby ďalším náhodným číslom. V tomto prípade je to jediný fragment s číslom 1. Náhodný faktor označme x. Skonštruujme nové čiastkové podpisy, ktoré budú mať nasledovný tvar

$$\begin{aligned} s'_{1a} &= h_1^{d-2x} \bmod N \\ s'_{1b} &= h_1^{d-x} \bmod N \end{aligned}$$

$$
\begin{aligned}
s'_{1c} &= h_1^d \bmod N \\
s'_2 &= h_2^d h_1^x \bmod N \\
s'_3 &= h_3^d h_1^x \bmod N
\end{aligned}
$$

Existujú dva druhy neželaných situácií. Prvou je nedodržanie prístupovej štruktúry, teda pokus extrahovať niektoré z fragmentov 2 a 3 bez extrahovania fragmentu s číslom 1. Druhou neželanou situáciou je, keď vlastník dokumentu dodrží pravidlá prístupovej štruktúry, avšak vyberie zlého reprezentanta čiastkového podpisu fragmentu, ktorý má aj nepovinné väzby. Túto situáciu vyriešime tak, že voľbu bude robiť automaticky algoritmus `Extract` a pre používateľa nebude viditeľná.

Prvý	Druhý	Tretí	Súčin	Overiteľný?
s'_{1a}			h_1^{d-2x}	Nie
s'_{1b}			h_1^{d-x}	Nie
s'_{1c}			h_1^d	Áno
	s'_2		$h_1^x h_2^d$	Nie
		s'_3	$h_1^x h_3^d$	Nie
s'_{1a}	s'_2		$h_1^{d-x} h_2^d$	Nie
s'_{1b}	s'_2		$h_1^d h_2^d$	Áno
s'_{1c}	s'_2		$h_1^{d+x} h_2^d$	Nie
s'_{1a}		s'_3	$h_1^{d-x} h_3^d$	Nie
s'_{1b}		s'_3	$h_1^d h_3^d$	Áno
s'_{1c}		s'_3	$h_1^{d+x} h_3^d$	Nie
	s'_2	s'_3	$h_1^{2x} h_2^d h_3^d$	Nie
s'_{1a}	s'_2	s'_3	$h_1^d h_2^d h_3^d$	Áno
s'_{1b}	s'_2	s'_3	$h_1^{d+x} h_2^d h_3^d$	Nie
s'_{1c}	s'_2	s'_3	$h_1^{d+2x} h_2^d h_3^d$	Nie

Obr. 5.4: Prehľad možných kombinácií čiastkových podpisov

Takto zostrojené čiastkové podpisy je možné zneužiť nasledovným útokom. Nech

$$
s_{new} = \left(\frac{s'_{1a}}{s'_{1b}}\right)^2 s'_2 s'_3 = (h_1^{d-2x-d+x})^2 h_1^x h_2^d h_1^x h_3^d = h_2^d h_3^d
$$

Podpis s_{new} je platným podpisom pre dokument pozostávajúci z fragmentov 2 a 3, čo je v rozpore s danou prístupovou štruktúrou. Skonštruujme nové čiastkové podpisy nasledovne

$$
s'_{1a} = h_1^{d-2y} \bmod N
$$

$$
\begin{aligned}
s'_{1b} &= h_1^{d-x} \bmod N \\
s'_{1c} &= h_1^{d} \bmod N \\
s'_{2a} &= h_2^{d} h_1^{x} \bmod N \\
s'_{2b} &= h_2^{d} h_1^{y} \bmod N \\
s'_{3a} &= h_3^{d} h_1^{x} \bmod N \\
s'_{3b} &= h_3^{d} h_1^{y} \bmod N
\end{aligned}
$$

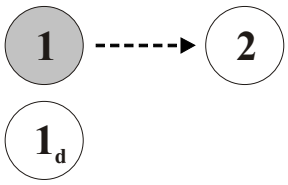
1.	2.	3.	Súčin	OK?	1.	2.	3.	Súčin	OK?
s'_{1a}			h_1^{d-2y}	Nie		s'_{2a}	s'_{3a}	$h_1^{2x} h_2^{d} h_3^{d}$	Nie
s'_{1b}			h_1^{d-x}	Nie		s'_{2a}	s'_{3b}	$h_1^{x+y} h_2^{d} h_3^{d}$	Nie
s'_{1c}			h_1^{d}	Áno		s'_{2b}	s'_{3a}	$h_1^{y+x} h_2^{d} h_3^{d}$	Nie
	s'_{2a}		$h_1^{x} h_2^{d}$	Nie		s'_{2b}	s'_{3b}	$h_1^{2y} h_2^{d} h_3^{d}$	Nie
	s'_{2b}		$h_1^{y} h_2^{d}$	Nie	s'_{1a}	s'_{2a}	s'_{3a}	$h_1^{d-2y+2x} h_2^{d} h_3^{d}$	Nie
		s'_{3a}	$h_1^{x} h_3^{d}$	Nie	s'_{1b}	s'_{2a}	s'_{3a}	$h_1^{d+x} h_2^{d} h_3^{d}$	Nie
		s'_{3b}	$h_1^{y} h_3^{d}$	Nie	s'_{1c}	s'_{2a}	s'_{3a}	$h_1^{d+2x} h_2^{d} h_3^{d}$	Nie
s'_{1a}	s'_{2a}		$h_1^{d-2y+x} h_2^{d}$	Nie	s'_{1a}	s'_{2a}	s'_{3b}	$h_1^{d-y+x} h_2^{d} h_3^{d}$	Nie
s'_{1b}	s'_{2a}		$h_1^{d} h_2^{d}$	Áno	s'_{1b}	s'_{2a}	s'_{3b}	$h_1^{d+y} h_2^{d} h_3^{d}$	Nie
s'_{1c}	s'_{2a}		$h_1^{d+x} h_2^{d}$	Nie	s'_{1c}	s'_{2a}	s'_{3b}	$h_1^{d+x+y} h_2^{d} h_3^{d}$	Nie
s'_{1a}	s'_{2b}		$h_1^{d-y} h_2^{d}$	Nie	s'_{1a}	s'_{2b}	s'_{3a}	$h_1^{d-y+x} h_2^{d} h_3^{d}$	Nie
s'_{1b}	s'_{2b}		$h_1^{d-x+y} h_2^{d}$	Nie	s'_{1b}	s'_{2b}	s'_{3a}	$h_1^{d+y} h_2^{d} h_3^{d}$	Nie
s'_{1c}	s'_{2b}		$h_1^{d+y} h_2^{d}$	Nie	s'_{1c}	s'_{2b}	s'_{3a}	$h_1^{d+x+y} h_2^{d} h_3^{d}$	Nie
s'_{1a}		s'_{3a}	$h_1^{d-2y+x} h_3^{d}$	Nie	s'_{1a}	s'_{2b}	s'_{3b}	$h_1^{d} h_2^{d} h_3^{d}$	Áno
s'_{1b}		s'_{3a}	$h_1^{d} h_3^{d}$	Áno	s'_{1b}	s'_{2b}	s'_{3b}	$h_1^{d-x+2y} h_2^{d} h_3^{d}$	Nie
s'_{1c}		s'_{3a}	$h_1^{d+x} h_3^{d}$	Nie	s'_{1c}	s'_{2b}	s'_{3b}	$h_1^{d+2y} h_2^{d} h_3^{d}$	Nie
s'_{1a}		s'_{3b}	$h_1^{d-y} h_3^{d}$	Nie					
s'_{1b}		s'_{3b}	$h_1^{d-x+y} h_3^{d}$	Nie					
s'_{1c}		s'_{3b}	$h_1^{d+y} h_3^{d}$	Nie					

Obr. 5.5: Prehľad možných kombinácií čiastkových podpisov

Touto konštrukciou sme vyriešili jednu časť problému. Rôzne alternatívy čiastkového podpisu prvého fragmentu používajú rôzne náhodné faktory. Tým sme zabezpečili odolnosť voči spomenutému útoku v prípade, že pri ňom nepoužijeme pôvodný čiastkový podpis (v tejto konštrukcii označený ako s'_{1c}). Aby sme úplne predišli možnému útoku musíme upraviť aj pôvodný čiastkový podpis. Aj túto alternatívu teda upravíme použitím ďalšieho náhodného faktora.

Ku každému fragmentu v prístupovej štruktúre, z ktorého vychádza aspoň jedna nepovinná väzba zostrojíme nový fragment v pôvodnom dokumente (budeme ho volať dummy-fragment). Dummy-fragment nebude mať žiadny obsah; bude mať iba svoje poradové číslo a náhodný faktor. V prípade, keď spolu s daným fragmentom nechceme extrahovať žiadne fragmenty viazané nepovinnou väzbou, extrahujeme jeho dummy-fragment.

Novú konštrukciu si ukážeme na zjednodušenom príklade (kvôli veľkému počtu kombinácií). Uvažujme štruktúru, ktorá obsahuje fragment 1 (prvej úrovne) previazaný s fragmentom 2 (druhej úrovne) nepovinnou väzbou. Navyše obsahuje dummy-fragment prislúchajúci k fragmentu 1.



Obr. 5.6: Príklad 3 - nepovinná väzba a dummy-fragment

Predpokladajme, že čiastkové podpisy prislúchajúce týmto fragmentom v pôvodnej schéme sú

$$\begin{aligned} s_1 &= h_1^d \bmod N \\ s_2 &= h_2^d \bmod N \\ s_D &= h_D^d \bmod N \end{aligned}$$

Čiastkový podpis dummy-fragmentu je označený symbolom s_D. Náhodný faktor fragmentu 1 označme x, náhodný faktor dummy-fragmentu označme q. Skonštruujme nové čiastkové podpisy nasledovne

$$\begin{aligned} s'_{1a} &= h_1^{d-x} \bmod N \\ s'_{1b} &= h_1^{d-q} \bmod N \\ s'_2 &= h_2^d h_1^x \bmod N \\ s'_D &= h_D^d h_1^q \bmod N \end{aligned}$$

Prvý	Druhý	Tretí	Súčin	Overiteľný?
s'_{1a}			h_1^{d-x}	Nie
s'_{1b}			h_1^{d-q}	Nie
	s'_2		$h_1^x h_2^d$	Nie
		s'_D	$h_1^q h_D^d$	Nie
s'_{1a}	s'_2		$h_1^d h_2^d$	Áno
s'_{1b}	s'_2		$h_1^{d-q+x} h_2^d$	Nie
s'_{1a}		s'_D	$h_1^{d-x+q} h_D^d$	Nie
s'_{1b}		s'_D	$h_1^d h_D^d$	Áno
	s'_2	s'_D	$h_1^{x+q} h_2^d h_D^d$	Nie
s'_{1a}	s'_2	s'_D	$h_1^{d+q} h_2^d h_D^d$	Nie
s'_{1b}	s'_2	s'_D	$h_1^{d+x} h_2^d h_D^d$	Nie

Obr. 5.7: Prehľad možných kombinácií čiastkových podpisov

5.3 Implementácia

Popíšme jednotlivé modifikované algoritmy. Algoritmus `Keygen` ostáva rovnaký ako pri pôvodnej schéme RSAProd. Algoritmus `Sign` pri svojej činnosti používa dvojrozmerné dynamické pole p na ukladanie alternatív jednotlivých čiastkových podpisov. Prvý rozmer je $n+m$ (m je počet dummy-fragmentov), druhý je premenlivý.

```
procedure Sign(M, CEAS, SK)
begin
   (1) vypočítame alebo náhodne zvolíme CES-tag T
   (2) zostrojíme dummy-fragmenty
   (3) h_i := H(T|i|M[i]) pre i ∈ {1,...,n+m}
   (4) Prepare(CEAS)
   (5) s_FULL := (CEAS|T|p[1,1]|...|p[n+m,max])
end
```

Na začiatku zvolíme CES-tag T. V druhom kroku zostrojíme dummy-fragmenty pre každý fragment, z ktorého vychádza aspoň jedna nepovinná väzba. Označíme ich číslami $n+1$ až $n+m$. V treťom kroku vypočítame hodnoty h_i podobne ako v pôvodnej schéme. Do vstupu pre hašovaciu funkciu H však nezahrnieme fyzickú reprezentáciu prístupovej štruktúry. Okrem hodnôt $h_1, \ldots, h_n$ vypočítame aj hodnoty $h_{n+1}, \ldots, h_{n+m}$ pre dummy-fragmenty. Ďalším krokom je volanie procedúry `Prepare`.

V poslednom kroku vytvoríme podpis pozostávajúci z fyzickej reprezentácie prístupovej štruktúry, CES-tagu T a všetkých alternatív čiastkových podpisov. Číslo max je index poslednej alternatívy čiastkového podpisu fragmentu s číslom $n+m$. Čiastkové podpisy a ich alternatívy zreťazíme v takom poradí, aby bolo neskôr možné určiť ich pôvodné indexy v poli p.

```
procedure Prepare(CEAS)
begin
   ( 1) p[i,1] := h_i^d pre i ∈ {1,...,n+m}
   ( 2) G := grafová reprezentácia CEAS
   ( 3) vrcholom v_i ∈ V(G) priraďme náhodné číslo: f_i
   ( 4) D := množina dummy-fragmentov
   ( 5) fragmentom d_j ∈ D priraďme náhodné číslo: f_j
   ( 6) while (E_1(G) ≠ ∅)
   ( 7) begin
   ( 8) . nájdime hranu r_i = (v_j, v_i) ∈ E_1(G) vchádzajúcu do v_i
   ( 9) . p[i,1] := p[i,1] * h_j^{f_i}
   (10) . p[j,1] := p[j,1] * h_j^{-f_i}
   (11) . odstráňme z grafu G hranu r_i
   (12) end
   (13) W(G) := {v_i ∈ V(G); ∃v_j ∈ V(G) : (v_i, v_j) ∈ E_2(G)}
   (14) vrcholom v_i ∈ W(G) priraďme očíslovaný zoznam:
   (15) . L_i = {f_j; ∃j : (v_i, v_j) ∈ E_2(G)}
   (16) while (V(G) ≠ ∅)
   (17) begin
   (18) . nájdime ľubovoľný list grafu G - vrchol v_i
   (19) . if (L_i ≠ ∅)
   (20) . . for f_l ∈ L_i: p[i, e_l + 1] := p[i,1] * h_i^{-e_l * f_l}
   (21) . . d_k := dummy-fragment fragmentu i
   (22) . . p[i,1] := p[i,1] * h_i^{-f_k}
   (23) . . p[k,1] := p[k,1] * h_i^{f_k}
   (24) . if (existuje hrana r_i = (v_j, v_i) vchádzajúca do v_i)
   (25) . . Y := {k; p[i,k] ≠ 0}
   (26) . . y := |Y|
   (27) . . for f_l ∈ L_j:
   (28) . . . for k ∈ Y: p[i, (e_l * y) + k] := p[i,k] * h_j^{f_l}
   (29) . . odstráňme z grafu G hranu r_i
   (30) . odstráňme z grafu G vrchol v_i
   (31) end
end
```

Procedúra `Prepare` pracuje s dvojrozmerným dynamickým poľom p. Prvok poľa $p[i,k]$ zodpovedá k-tej alternatíve čiastkového podpisu fragmentu s číslom i. Predpokladajme, že prvky poľa sa inicializujú (na nulu) pri prvom použití. G je grafová reprezentácia prístupovej štruktúry, $V(G)$ je množina vrcholov grafu, D je množina dummy-fragmentov. $E_1(G)$ je množina hrán zodpovedajúca povinným väzbám, $E_2(G)$ je množina hrán zodpovedajúca nepovinným väzbám. $W(G)$ je množina tých vrcholov, v ktorých začína aspoň jedna hrana zodpovedajúca nepovinnej väzbe. Číslo f_i je náhodný faktor priradený každému vrcholu a dummy-fragmentom. Každému vrcholu $v_i \in W(G)$ je priradený zoznam L_i, ktorý obsahuje náhodné faktory všetkých vrcholov, ktoré sú previazané s vrcholom v_i nepovinnou väzbou. Prvky zoznamu sú očíslované, začínajúc indexom 1. Poradové číslo prvku $f_l \in L_i$ označíme e_l.

Po inicializácii poľa p `(1)` a vygenerovaní náhodných faktorov f_i `(3, 5)` algoritmus zakomponuje do čiastkových podpisov povinné väzby `(6-12)`. Postupne prechádza všetky hrany z $E_1(G)$ a modifikuje čiastkové podpisy fragmentov na oboch koncoch väzby. Pritom využíva náhodný faktor fragmentu, v ktorom väzba končí. Pri tomto procese nevznikajú nové alternatívy podpisov. Po príslušnej modifikácii odstráni z grafu použitú hranu. Po odstránení všetkých hrán z $E_1(G)$ pokračuje algoritmus spracovaním nepovinných väzieb `(13-31)`.

Najprv vytvorí množinu $W(G)$ `(13)` a ku každému vrcholu v_i z tejto množiny vytvorí zoznam L_i `(14-15)`. Prechádza všetkými vrcholmi smerom od listov ku koreňu. Ak zoznam L_i práve navštíveného vrcholu v_i je neprázdny, vytvorí nové alternatívy čiastkového podpisu fragmentu i `(20)`. Následne modifikuje prvú alternatívu podpisu náhodným faktorom príslušného dummy-fragmentu `(21-23)`.

Ak vo vrchole v_i končí hrana $r_i = (v_j, v_i)$, vytvoríme toľko kópií každého čiastkového podpisu fragmentu i, koľko prvkov obsahuje zoznam L_j. Tieto kópie postupne modifikujeme príslušnými náhodnými faktormi zo zoznamu L_j `(27-28)`. Hranu r_i a vrchol v_i z grafu odstránime. Ak už graf neobsahuje žiadny vrchol, algoritmus končí.

Na ilustráciu výstupu procedúry `Prepare` použijeme prístupovú štruktúru z obrázku 4.1. Nasledujúce tabuľky zodpovedajú poľu p tejto štruktúry. Prvá tabuľka obsahuje čiastkové podpisy dummy-fragmentov, druhá tabuľka podpisy obyčajných fragmentov.

9	10	11
$h_9^d h_1^{f_9}$	$h_{10}^d h_3^{f_{10}}$	$h_{11}^d h_4^{f_{11}}$

Obr. 5.8: Výstup procedúry `Prepare` - 1. časť

1	2	3	4	5	6	7	8
$h_1^{d-f_3-f_9}$	h_2^d	$h_3^{d-f_5-f_{10}}h_1^{f_3}$	$h_4^{d-f_8-f_{11}}$	$h_5^d h_3^{f_5}$	$h_6^d h_3^{f_6}$	$h_7^d h_4^{f_7}$	$h_8^d h_4^{f_8}$
$h_1^{d-f_3-f_4}$	$\emptyset$	$h_3^{d-f_5-f_6}h_1^{f_3}$	$h_4^{d-f_8-f_7}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$\emptyset$	$\emptyset$	$\emptyset$	$h_4^{d-f_8-f_{11}}h_1^{f_4}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$\emptyset$	$\emptyset$	$\emptyset$	$h_4^{d-f_8-f_7}h_1^{f_4}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

Obr. 5.9: Výstup procedúry `Prepare` - 2. časť

Tretím algoritmom v pôvodnej schéme je algoritmus `Extract`. Tento algoritmus je v novej schéme zmenený na dvoch miestach. V treťom kroku pri konštrukcii výsledného extrahovaného podpisu nie je oproti pôvodnému riešeniu zahrnutá fyzická reprezentácia prístupovej štruktúry, pretože nie je potrebná pre overovací algoritmus.

V druhom kroku je navyše nutné vybrať správnu alternatívu čiastkového podpisu. Uvažujme vrchol v na najnižšej úrovni (list). Nech tento vrchol je viazaný nepovinnou väzbou z vrchola u. Nech m je počet vrcholov previazaných s vrcholom u nepovinnou väzbou, ktoré chceme extrahovať. V tom prípade vyberieme $(m+1)$-vú alternatívu podpisu.

Uvažujme teraz vnútorný vrchol u. Opäť vyberáme $(m+1)$-vú alternatívu čiastkového podpisu. Ak $m = 0$, tak navyše extrahujeme aj dummy-fragment prislúchajúci k vrcholu u (zaradíme ho do množiny X). Ak do vrcholu u smeruje nepovinná väzba z vrcholu vyššej úrovne, tak $(m+1)$-vá alternatíva je množinou alternatív. Vhodnú alternatívu z tejto množiny vyberieme rovnakým spôsobom, aký sme použili pri vrchole v (resp. ľubovoľnom liste). Rovnaký postup aplikujeme na vnútorné vrcholy v ľubovoľnej úrovni.

`procedure Extract(`M`,` s_{FULL}`,` X`,` PK`)`
`begin`
`(1) rozdeľme` s_{FULL}`:` $s_{FULL} = (CEAS|T|p[1,1]|\ldots|p[n+m,max])$
`(2)` s_f `:=` $\prod_{i \in X} p[i,k_i]$ `mod` N `(volíme vhodné alternatívy)`
`(3)` s_{EXT} `:=` $T|s_f$
`end`

Posledným algoritmom schémy je algoritmus `Verify`. Zmeny v procedúre spočívajú iba vo vynechaní niektorých krokov. Konkrétne boli vynechané všetky časti týkajúce sa prístupovej štruktúry, ktorá v tejto fáze schémy vôbec nevystupuje. Nie je dodaná so vstupom ako časť extrahovaného podpisu, nie je nutná na počítanie hodnôt h_i a nakoniec nie je potrebná ani pri záverečnom teste ako jedna časť konjunkcie. Dosiahli sme teda stanovený cieľ

odstrániť prístupovú štruktúru v komunikácii vlastníka dokumentu s treťou stranou a pri overovaní extrahovaného podpisu.

```
procedure Verify(M', s_EXT, PK)
begin
   (1) X := c(M')
   (2) rozdeľme s_EXT na zložky: s_EXT = T|s_f
   (3) h_i := H(T|i|M'[i]) pre i ∈ X
   (4) if (s_f^e = ∏_{i∈X} h_i mod N)
   (5) . verifikácia prebehla úspešne
   (6) else
   (7) . verifikácia zlyhala
end
```

Kapitola 6

Záver

V práci sme zaviedli novú stromovú prístupovú štruktúru. Výhodou tejto štruktúry je možnosť definovať väzby medzi fragmentami a lineárna priestorová zložitosť. Ďalej sme v práci navrhli novú podpisovú schému, ktorá je modifikáciou schémy RSAProd. Hlavnou výhodou novej schémy je zmenšenie dĺžky extrahovaného podpisu na konštantnú veľkosť.

Idea spočíva v tom, že odstránime nutnosť prítomnosti prístupovej štruktúry pri overovaní extrahovaného podpisu. Pravidlá, ktoré táto štruktúra stanovuje (a teda v podstate celú prístupovú štruktúru) vložíme už do samotného pôvodného podpisu konštruovaného autorom dokumentu. Vlastník dokumentu následne nebude schopný vytvoriť extrahovaný podpis verifikovateľný treťou stranou, ktorý nespĺňa pravidlá prístupovej štruktúry.

Rozvinutie tejto myšlienky a hľadanie jej ďalšieho využitia môže byť námetom na ďalšiu prácu v tejto oblasti. Jednou možnosťou je rozšíriť stromovú prístupovú štruktúru o nové typy väzieb – napríklad väzby, ktoré spájajú viac ako dva fragmenty. Čiastočne sme túto myšlienku už využili pri definovaní dummy-fragmentov. Ak spolu s daným fragmentom nie sú extrahované žiadne fragmenty viazané nepovinnou väzbou, extrahujeme dummy-fragment. Túto vlastnosť môžeme chápať ako väzbu medzi daným fragmentom, všetkými fragmentami, ktoré sú s ním viazané nepovinnou väzbou a dummy-fragmentom.

Ďalším námetom je vytvorenie takej modifikácie novej schémy, ktorá by vedela pracovať aj s inou ako stromovou prístupovou štruktúrou.

Literatúra

[1] L. Bull, D. McG Squire, and Y. Zheng. A hierarchical extraction policy for content extraction signatures. Technical Report 2003/147, School of Computer Science and Software Engineering, Monash University, Australia 3800, December 2003.

[2] Robert Johnson, David Molnar, Dawn Xiaodong Song, and David Wagner. Homomorphic signature schemes. In *CT-RSA*, pages 244–262, 2002.

[3] Ron Steinfeld, Laurence Bull, and Yuliang Zheng. Content extraction signatures. *Lecture Notes in Computer Science*, 2288:285–??, 2002.