

# Podpisy založené na hašovacích funkciách

Martin Stanek

Department of Computer Science  
Comenius University  
`stanek@dcs.fmph.uniba.sk`

Cryptology 1 (2020/21)

# Content

Lamportova schéma  
vylepšenia

WOTS a WOTS+

Merkleho hašovacie stromy  
Merkleho podpisová schéma (MSS)

XMSS

Bezstavové schémy

- ▶ podpisové schémy odolné voči kvantovým výpočtom
  - ▶ bezpečnosť nezávisí na zložitosti faktorizácie, DL a pod.
  - ▶ obvykle hašovacie funkcie - odolnosť vzoru, odolnosť druhého vzoru, odolnosť voči kolíziám
- ▶ ukážeme niektoré schémy a ich obmedzenia
  - ▶ počet použití
  - ▶ stav
  - ▶ veľkosť kľúčov, resp. podpisov

# Lamportova schéma

- ▶ Lamport (1979)
- ▶  $f : X \rightarrow Y$  (napr. hašovacia funkcia)
- ▶  $n$  - dĺžka podpisovanej správy (príp. jej odtačku)
- ▶ súkromný kľúč:  $x_{i,j} \xleftarrow{\$} X$  pre  $i = 1, \dots, n, j \in \{0, 1\}$
- ▶ verejný kľúč:  $y_{i,j} = f(x_{i,j})$
- ▶ správa na podpis  $m = m_1, m_2, \dots, m_n$  (príp. jej odtlačok)
- ▶ podpis:  $\sigma = (x_{1,m_1}, x_{2,m_2}, \dots, x_{n,m_n})$
- ▶ overenie podpisu  $\sigma = (\sigma_1, \dots, \sigma_n)$  pre správu  $m$ :

$$f(\sigma_i) \stackrel{?}{=} y_{i,m_i} \quad i = 1, \dots, n.$$

# Lamportova schéma - poznámky

- ▶ schéma je **jednorazová**
  - ▶ podpísanie dvoch správ (pokiaľ sa tieto nelíšia len v 1 bite) umožňuje skombinovať korektný podpis pre novú správu
  - ▶ podpisovanie odtlačku správy veľmi nepomôže, stačí získať podpisy správ, ktorých odtlačky pokryjú výskyt 0 a 1 na (skoro) všetkých pozíciách ( $O(\lg n)$  správ bude stačiť)
- ▶ veľkosť kľúčov (nech  $f$  je 256-bitová hašovacia funkcia a  $n = 256$ ):
  - ▶ verejný kľúč:  $2 \cdot 256 \cdot 256 = 16$  KiB
  - ▶ súkromný kľúč: 16 KiB (pre 256-bitové hodnoty  $x_{i,j}$ )
- ▶ veľkosť podpisu: 8 KiB
- ▶ rýchlosť nie je problém

# Lamportova schéma - vylepšenia (1)

- ▶ efektivita, Merkle (1979)
- ▶ skrátenie súkromného kľúča
  - ▶ generovanie hodnôt pomocou PRNG, resp. PRF
- ▶ skrátenie verejného kľúča
  - ▶  $y = H(y_{1,0}, y_{1,1}, \dots, y_{n,0}, y_{n,1})$
  - ▶ hodnoty  $y_{i,1-m_i}$  (teda aj pri podpise nevyužité hodnoty) uviesť spolu s podpisom (podpis 2 krát dlhší ako predtým)
  - ▶ overenie zahŕňa výpočet hodnôt  $y_{i,m_i}$  a následne verifikáciu  $y$

## Lamportova schéma - vylepšenia (2)

- ▶ skrátenie kľúčov a podpisov na približne polovicu:
  - ▶ k podpisovanej správe (k odtlačku) pridáme  $\lfloor \lg n \rfloor + 1$  bitov, počítadlo 0:  
 $m' = m || (\#_0 m)_2$
  - ▶ nech  $n' = n + \lfloor \lg n \rfloor + 1$
  - ▶ súkromný kľúč:  $x_i \xleftarrow{\$} X$  pre  $i = 1, \dots, n'$
  - ▶ verejný kľúč:  $y_i = f(x_i)$
  - ▶ podpis správy je postupnosť:

$$(x_i)_{i \in I} \quad \text{kde } I = \{1 \leq i \leq n' \mid m'_i = 1\}$$

- ▶ v každom  $m'$  je aspoň jedna 1
- ▶ zmena 0 na 1 v  $m$  - príslušné  $x_i$  nie je v podpise
- ▶ zmena 1 na 0 v  $m$  - väčší počet 0, teda v kontrolnom súčte sa aspoň jedna 0 zmení na 1, príslušné  $x_i$  nie je v podpise

- ▶ WOTS - Winternitz one time signature
- ▶ skrátenie podpisov a zvýšenie časovej zložitosti (TMTO)
- ▶ funkcia  $f : X \times X \rightarrow X$  parametrizovaná kľúčom (akoby MAC)
  - ▶ označenie:  $f(k, x) = f_k(x)$
- ▶ iterovanie funkcie  $f$ :
  - ▶  $f_k^0(x) = k, f_k^1(x) = f_k(x), f_k^2(x) = f_{f_k(x)}(x)$
  - ▶ všeobecne  $f_k^r(x) = f_{f_k^{r-1}(x)}(x)$
  - ▶ iný varianty WOTS: priamočiara iterácia (hašovacej) funkcie – vyžaduje silnejší predpoklad (odolnosť voči kolíziám)
- ▶ parameter  $w > 1$ , napr.  $w = 16$  alebo  $w = 32$
- ▶ správu  $m$  vyjadríme vo  $w$ -árnej sústave
  - ▶  $m = (m_1, \dots, m_{l_1})$ , kde  $0 \leq m_i < w$  pre  $i = 1, \dots, l_1$
- ▶ kontrolný súčet správy:  $C = \sum_{i=1}^{l_1} (w - 1 - m_i)$
- ▶ nech  $l = l_1 + l_2$ , kde  $l_2$  je maximálna dĺžka  $C$  ( $w$ -árna sústava)

## WOTS (2)

- ▶ súkromný kľúč:  $k_1, \dots, k_l \xleftarrow{\$} X$
- ▶ verejný kľúč:  $(x, y_1, \dots, y_l)$ , kde
  - ▶  $x \xleftarrow{\$} X$
  - ▶  $y_i = f_{k_i}^{w-1}(x)$  pre  $i = 1, \dots, l$
- ▶ podpisovanie:
  1. správu a jej kontrolný súčet rozdelíme na bloky:  $m \parallel C \mapsto m_1, \dots, m_l$
  2. podpis  $\sigma = (\sigma_1, \dots, \sigma_l) = (f_{k_1}^{m_1}(x), \dots, f_{k_l}^{m_l}(x))$
- ▶ overovanie podpisu:
  1. pre dané  $m$  vypočítame kontrolný súčet a rozdelíme na bloky
  2. overíme:

$$f_{\sigma_i}^{w-1-m_i}(x) \stackrel{?}{=} y_i \quad i = 1, \dots, l$$

## WOTS (3)

- ▶ korektnosť schémy triviálne
- ▶ bez kontrolného súčtu schéma nie je bezpečná
  - ▶ z hodnoty  $\sigma_i$  vie ktokoľvek pre  $m'_i \geq m_i$  počítať  $\sigma'_i = f_{k_i}^{m'_i}(x) = f_{\sigma_i}^{(m'_i - m_i)}(x)$
  - ▶ teda korektný podpis pre  $i$ . blok
  - ▶ kontrolný súčet chráni pred takýmito posunmi
- ▶ stále jednorazová schéma

# WOTS - konkrétne parametre

- ▶ veľkosť kľúčov (nech  $f$  je 256-bitová hašovacia funkcia a  $|m| = 256$ ):
  - ▶ pre  $w = 16$ :  $l_1 = 256/4 = 64$ ,  $l_2 = \lfloor \lg(64 \cdot 15)/4 \rfloor + 1 = 2$
  - ▶ teda  $l = 66$
  - ▶ verejný kľúč:  $(l + 1) \cdot 256 \approx 2.1$  KiB
  - ▶ súkromný kľúč:  $l \cdot 256 \approx 2.1$  KiB (pre 256 bitové hodnoty)
- ▶ veľkosť podpisu:  $l \cdot 256 \approx 2.1$  KiB
  - ▶ cca.  $\lg w = 4$  krát kratšie ako v pôvodnej Lamportovej schéme
- ▶ rýchlosť – porovnanie s pôvodnou Lamportovou schémou
  - ▶ podpisovanie: cca.  $l \cdot w/2$  výpočtov  $f$  vs. 0
  - ▶ overovanie: cca.  $l \cdot w/2$  výpočtov  $f$  vs.  $|m|$  výpočtov  $f$  (WOTS približne  $w/(2 \cdot \lg w)$  krát pomalšie)

- ▶ Hülising, 2013
- ▶ podobná schéma ako WOTS, iná iterácia funkcie  $f$
- ▶ tesnejší dôkaz bezpečnosti ako pre WOTS
  - ▶ slabšie alebo štandardnejšie predpoklady na vlastnosti  $f$
  - ▶ WOTS<sup>+</sup> môže nahradiť WOTS v iných konštrukciách
- ▶ iterovanie funkcie  $f : K \times X \rightarrow X$ 
  - ▶ vstupy: kľúč  $k \in K$ ,  $x \in X$ , počítadlo  $i \in \mathbb{N}$ , znáhodňujúce prvky  $\mathbf{r} = (r_1, \dots, r_j)$  pre  $j \geq i$
  - ▶ výpočet:

$$c_k^0(x, \mathbf{r}) = x$$

$$c_k^1(x, \mathbf{r}) = f_k(c_k^0(x, \mathbf{r}) \oplus r_1)$$

...

$$c_k^i(x, \mathbf{r}) = f_k(c_k^{i-1}(x, \mathbf{r}) \oplus r_i)$$

## WOTS<sup>+</sup> (2)

- ▶ parametre  $w, l = l_1 + l_2$  ako pri WOTS
- ▶ súkromný kľúč:  $x_1, \dots, x_l \xleftarrow{\$} X$
- ▶ verejný kľúč:  $((\mathbf{r}, k), y_1, \dots, y_l)$ 
  - ▶  $k \xleftarrow{\$} K$
  - ▶  $\mathbf{r} = (r_1, \dots, r_{w-1})$ , kde  $r_i \xleftarrow{\$} X$
  - ▶  $y_i = c_k^{w-1}(x_i, \mathbf{r})$ , pre  $i = 1, \dots, l$
- ▶ podpisovanie:
  1. (zhodné s WOTS) správu a jej kontrolný súčet rozdelíme na bloky:  
 $m \parallel C \mapsto m_1, \dots, m_l$
  2. podpis  $\sigma = (\sigma_1, \dots, \sigma_l) = (c_k^{m_1}(x_1, \mathbf{r}), \dots, c_k^{m_l}(x_l, \mathbf{r}))$

*Poznámka:* kontrolný súčet (rovnako ako pri WOTS) zabezpečuje, že pre dané  $m_1, \dots, m_l$  ľubovoľná iná správa obsahuje aspoň jedno  $m'_j < m_j$

## WOTS<sup>+</sup> (3)

### ► overovanie:

1. pre dané  $m$  vypočítame kontrolný súčet a rozdelíme na bloky
2. overíme:

$$c_k^{w-1-m_i}(\sigma_i, \mathbf{r}_{m_i+1, w-1}) \stackrel{?}{=} y_i \quad i = 1, \dots, l$$

kde  $\mathbf{r}_{m_i+1, w-1} = (r_{m_i+1}, \dots, r_{w-1})$

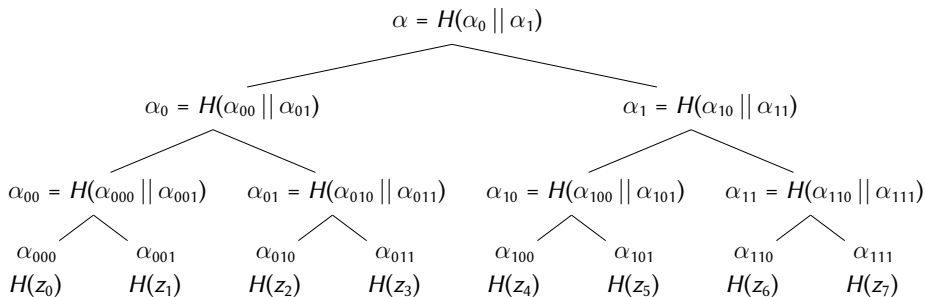
### ► skrátenie kľúčov:

- hodnoty  $y_i$  verejného kľúča vo WOTS a WOTS<sup>+</sup> sú dopočítané pri overovaní ... je možné ich nahradiť vo verejnom kľúči odtlačkom
- hodnoty v  $\mathbf{r}$  môžu byť generované zo seedu
- podobne pre hodnoty v súkromnom kľúči

# Merkleho hašovacie stromy

- ▶ Merkle, 1979
- ▶ riešenie problému s jediným použitím podpisovej schémy
- ▶ viacero jednorazových schém
  - ▶ skombinované do spoločnej stromovej štruktúry
- ▶ Merkleho hašovacie stromy – rôzne aplikácie, napr.
  - ▶ súborové systémy (napr. ZFS), BitTorrent, Bitcoin, Git, ...
- ▶ Konštrukcia (binárny strom):
  - ▶ vstup: dáta  $z_0, \dots, z_{2^h-1}$ , pre nejaké  $h \geq 1$
  - ▶ pre zjednodušenie predpokladajme  $2^h$  vstupných dát
  - ▶  $H$  – hašovacia funkcia
  - ▶ hodnoty listov stromu:  $H(z_i)$ , pre  $i = 0, \dots, 2^h - 1$
  - ▶ hodnota vrcholu  $v$ :  $H(a || b)$ , kde  $a$ , resp.  $b$ , je hodnota ľavého, resp. pravého, potomka vrchola  $v$

# Príklad



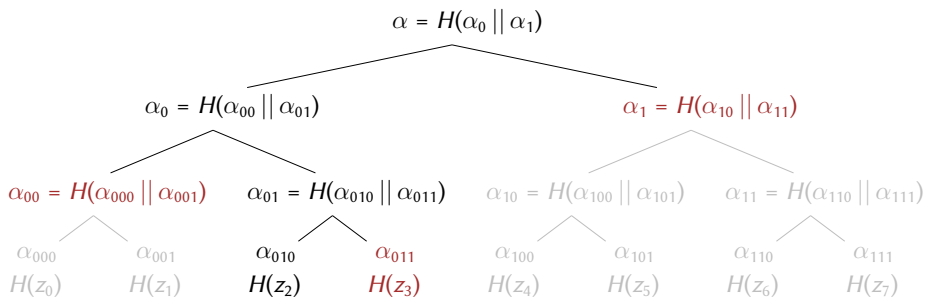
Merkleho hašovací strom pre  $h = 3$

# Merkleho podpisová schéma (MSS)

- ▶ dáta pre listy v strome - verejné kľúče OTS schém
- ▶ verejný kľúč: hodnota v koreni Merkleho hašovacieho stromu
- ▶ súkromný kľúč:
  - ▶ kľúč pre (vhodnú) konštrukciu, ktorou generujeme OTS schémy
  - ▶ príp. sada súkromných kľúčov OTS schém
- ▶ použitie (dokážeme podpísať  $2^h$  správ):
  - ▶ postupne používame jednotlivé OTS schémy
  - ▶ pamätáme si, koľko správ už bolo podpísaných
- ▶ podpis správy obsahuje:
  - ▶ podpis správy v práve použitej OTS schéme
  - ▶ verejný kľúč použitej OTS schémy
  - ▶ autentizačnú cestu

# Autentizačná cesta

- ▶ autentizačná cesta pre daný list stromu:
  - ▶ postupnosť hodnôt súrodeneckých vrcholov na ceste ku koreňu
  - ▶ umožňuje vypočítať hodnotu koreňa z hodnoty listu a ďalších  $h$  hodnôt



autentizačná cesta pre  $\alpha_{010}$ :  $\text{auth}(\alpha_{010}) = (\alpha_{011}, \alpha_{00}, \alpha_1)$

# Merkleho podpisová schéma - overenie podpisu

- ▶ vstupy:
  - ▶ verejný kľúč MSS (hodnota v koreni)
  - ▶ správa
  - ▶ podpis v OTS schéme a jej verejný kľúč
  - ▶ autentizačná cesta
- ▶ postup:
  1. overíme podpis správy v OTS schéme
  2. vypočítame hodnotu koreňa (z verejného kľúča OTS schémy a autentizačnej cesty)
  3. porovnáme získanú hodnotu s verejným kľúčom MSS
- ▶ bezpečnosť schémy závisí na vlastnostiach:
  - ▶  $H$  použitej v Merkleho hašovacom strome
  - ▶ použitej OTS schémy
  - ▶ kryptografických konštrukcií použitých pri generovaní OTS schém

# Merkleho podpisová schéma - poznámky

- ▶ pri generovaní schémy potrebné vygenerovať celý strom
  - ▶ obmedzuje prakticky použiteľnú veľkosť  $h$
- ▶ výpočet autentizačnej cesty:
  - ▶ Merkle tree traversal - sekvenčné generovanie listov a príslušných autentizačných ciest (využitie poradia používaných listov)
  - ▶ čas  $O(h)$ , pamäť  $O(h)$  (Szydło, 2004)
- ▶ Čo po vyčerpaní všetkých  $2^h$  OTS schém?
  - ▶ vygenerovať nový strom + distribúcia nového verejného kľúča
  - ▶ poslednou schémou podpísať koreň nového stromu
  - ▶ mať taký veľký strom, že situácia nenastane
- ▶ podpisy sú dlhšie kvôli autentizačnej ceste –  $h$  odtlačkov navyše
  - ▶ veľkosť stromu ovplyvňuje dĺžku podpisov

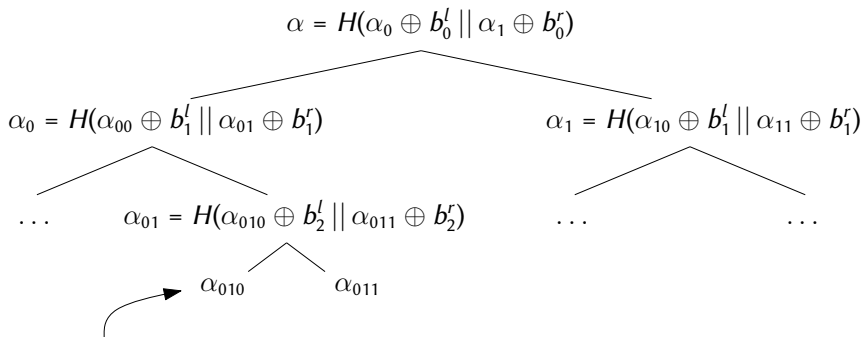
## Merkleho podpisová schéma - poznámky (2)

- ▶ použitie WOTS (WOTS<sup>+</sup>) v MSS
  - ▶ z podpisu sa dá dopočítať verejný kľúč OTS schémy (okrem  $x$  pre WOTS, resp.  $(r, k)$  pre WOTS<sup>+</sup>)
  - ▶ skrátenie podpisu, idea použitá napr. v XMSS
- ▶ problém: schéma nie je jednorazová, ale zrazu je **stavová**
  - ▶ pre bezpečnosť: aby sme OTS nepoužili viackrát
  - ▶ pre efektívnosť: výpočet autentizačnej cesty
- ▶ stav pre niektoré použitia nemusí vadiť
  - ▶ napr. CA a podpisovanie certifikátov v HSM
- ▶ vo všeobecnosti je stav problém
  - ▶ load-balancing, obnova systému zo zálohy (starý stav) a pod.

# XMSS (eXtended Merkle Signature Scheme)

- ▶ Buchmann, Dahmen, Hülsing (2011)
- ▶ modifikácia Merkleho stromov
  - ▶ xorovanie *masiek* pri agregácii hodnôt
  - ▶ v listoch *zavesené* tzv. L-stromy pre uloženie verejných kľúčov WOTS schém (spôsob hašovania verejného kľúča pre následné dôkazy bezpečnosti)
  - ▶ motivácia: stačí požadovať odolnosť druhého vzoru namiesto odolnosti voči kolíziám
- ▶ použitie WOTS – verejný kľúč nie je potrebné uvádzať v podpise
  - ▶ vypočíta sa z podpisu a overí spolu s overením XMSS stromu
- ▶ masky
  - ▶ pre každú nelistovú hĺbkú stromu zvolená ľavá a pravá maska
  - ▶ masky sú náhodne volené reťazce potrebnej dĺžky

## XMSS strom



koreň L-stromu pre verejný kľúč WOTS schémy  
(nová sada masiek, rovnaká pre všetky L-stromy)

# XMSS – poznámky

## ► L-strom

- rovnaká konštrukcia ako XMSS strom (nové nezávislé masky, ale rovnaké v každom L-strome)
- listy – hodnoty verejného kľúča WOTS ( $x, y_1, \dots, y_l$ )
- nemusí byť mocnina 2 – umiestnenie hodnôt vyššie (aby každý nelistový vrchol mal dvoch potomkov)

## ► stále si potrebujeme pamätať stav

## ► predstava o praktických parametroch (2011):

- $h = 20$ , teda kapacita na  $2^{20}$  podpisov
- použitie SHA-256, dĺžka verejného/súkromného kľúča: 13 568 / 280 bitov
- procesor i5 M-540, bezpečnosť je interpretovaná ako *bit security*

w	Podpis/Overenie	Generovanie	Podpis	Bezpečnosť
16	7.00 / 0.52 [ms]	466 [s]	22 296 bitov	196
64	15.17 / 1.02 [ms]	1 099 [s]	16 664 bitov	146

## ► novší návrh a parametre: RFC 8391 (2018)

# Bezstavové schémy

- ▶ odstránenie stavu z podpisovej schémy
- ▶ základná idea (Goldreich): obrovský binárny strom, napr.  $h = 256$ 
  - ▶ každý vrchol reprezentuje jednu OTS schému
  - ▶ schému vieme vygenerovať zo súkromného kľúča a pozície vrchola (PRF)
- ▶ súkromný kľúč: náhodný reťazec bitov
- ▶ verejný kľúč: verejný kľúč OTS schémy v koreni stromu ( $Y$ )

# Podpisovanie

- ▶ podpisovanie správy  $m$  (resp. jej odtlačku), napr. dĺžky 256 bitov:
  1. hodnota správy určí konkrétny list  $\beta$  v strome
  2. vypočítame parametre (kľúče) všetkých schém pre vrcholy (a ich súrodencov) na ceste od  $\beta$  ku koreňu (príslušné verejné kľúče označme  $Y_h^l, Y_h^r, \dots, Y_1^l, Y_1^r$ )
  3. správu podpíšeme OTS schémou pre  $\beta$  (označme  $\sigma_m$ )
  4. každú dvojicu verejných kľúčov podpíšeme príslušnou OTS schémou v rodičovskom vrchole (podpisy  $\sigma_1, \dots, \sigma_{h-1}$ )
  5. dvojicu podpísanú schémou v koreni označme  $\sigma_0$
  6. podpis správy:  $(\sigma_m, \sigma_0, \dots, \sigma_{h-1}, Y_1^l, Y_1^r, \dots, Y_h^l, Y_h^r)$

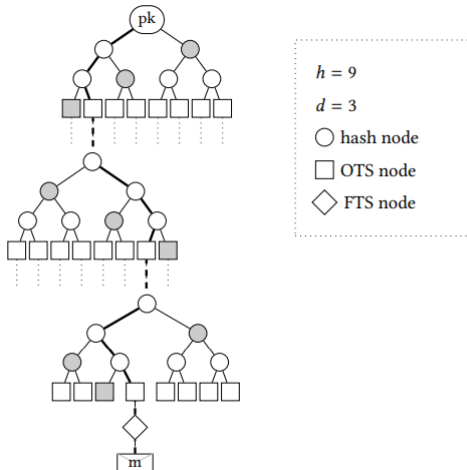
# Overenie podpisu

- ▶ vstup:
  - ▶ správa  $m$  (resp. odtlačok)
  - ▶ verejný kľúč z koreňa stromu:  $Y$
  - ▶ podpis  $(\sigma_m, \sigma_0, \dots, \sigma_{h-1}, Y_1^l, Y_1^r, \dots, Y_h^l, Y_h^r)$
- ▶ postup overenia:
  1. z  $m$  určíme konkrétny list  $\beta$
  2. overíme podpis  $\sigma_m$  (pomocou  $Y_h^l$  alebo  $Y_h^r$ )
  3. overíme *certifikačnú cestu*, teda podpisy  $(\sigma_1, \dots, \sigma_{h-1})$  pomocou verejných kľúčov z podpisu
  4. overíme podpis  $\sigma_0$  pomocou verejného kľúča  $Y$
- ▶ nepotrebujeme stav, lebo pravdepodobnosť kolízie je zanedbateľná
- ▶ podpisy vo vrcholoch certifikačnej cesty sú vždy rovnaké
  - ▶ certifikujú sa verejné kľúče potomkov
  - ▶ teda stačí OTS schéma
- ▶ nevýhoda: dlhé podpisy, nevhodné pre praktické použitie

# Optimalizácie bezstavovej schémy

- ▶ kombinácia viacerých techník
- ▶ deterministické/pseudonáhodné určenie listu v strome pre podpisovanú správu
- ▶ použite *few-time* podpisových schém namiesto WOTS v listových vrcholoch
  - ▶ len pre podpisovanie samotných správ
  - ▶ bez neželaných dopadov vieme podpísať zopár správ (povedzme 4)
  - ▶ zmenšenie stromu (potrebného počtu listových vrcholov)
  - ▶ príklady schém: HORS, HORST, FORS
- ▶ rozdelenie veľkého stromu na viacero úrovní menších Merkleho stromov

# Optimalizácie bezstavovej schémy (2)



**Figure 1: An illustration of a (small) SPHINCS structure.**

Zdroj: D. Bernstein et al.: The SPHINCS+ Signature Framework (2019)

# PQC štandardizácia

- ▶ v 2. kole bol jeden návrh tohto typu: SPHINCS+
- ▶ v 3. kole ako “alternate” (nie ako finalista)

NIST Status Report on the 2nd Round:

*SPHINCS+ provides very solid assurance of its security claims, but this comes at a substantial cost in performance—it is slower, and its signatures are considerably larger than most other signature schemes. ...*

*It is difficult to imagine TLS with SPHINCS+ as the signature algorithm providing acceptable performance. Both the speed and size of signatures would be unacceptable.*

- ▶ dĺžky v bajtoch (*s* – size-optimized (small), *f* – speed-optimized (fast)):

level	private key	public key	signature	
128	64	32	8 080	<i>SPHINCS+-128s</i>
128	64	32	16 976	<i>SPHINCS+-128f</i>
256	128	64	29 792	<i>SPHINCS+-256s</i>
256	128	64	49 216	<i>SPHINCS+-256f</i>

# Záver

- ▶ podpisové schémy založené na hašovacích funkciách
  - ▶ vždy obmedzenie na max. počet podpisov
  - ▶ princípy sú jednoduché
  - ▶ optimalizácie pre praktickú použiteľnosť a pre slabšie bezpečnostné predpoklady sú zložitejšie