CRYPTREC
Cryptography Research and Evaluation Committees

ハッシュベース署名FIPS205の構成の解説

廣瀬勝一
（福井大学）

2025年7月25日

NIST FIPS 205

- 2024年8月
- SLH-DSAを規定
  - StateLess Hash-based Digital Signature Algorithm
  - 状態を持たないハッシュベース署名方式
  - 署名に用いられる秘密鍵の使用・未使用の管理が不要
  - SPHINCS$^+$ ver. 3.1に基づく（ほぼ同じ）

【注意】従来のハッシュベース署名方式はすべてstateful
  - 署名に用いられる秘密鍵の使用・未使用の管理が必要

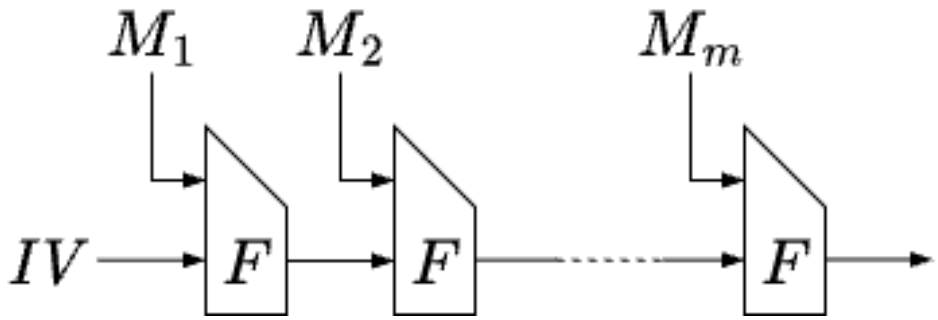
ハッシュベース署名方式

- 1回署名方式
  - Lamport [Lam79, DH76]
  - Winternitz [Mer87]
- マークル木を用いた署名方式 [Mer79]
- マークル木を用いた署名方式の階層構造による署名方式 [BCDDK06]
- 数回（few-time）署名方式 [RR02]

ハッシュ関数の構成

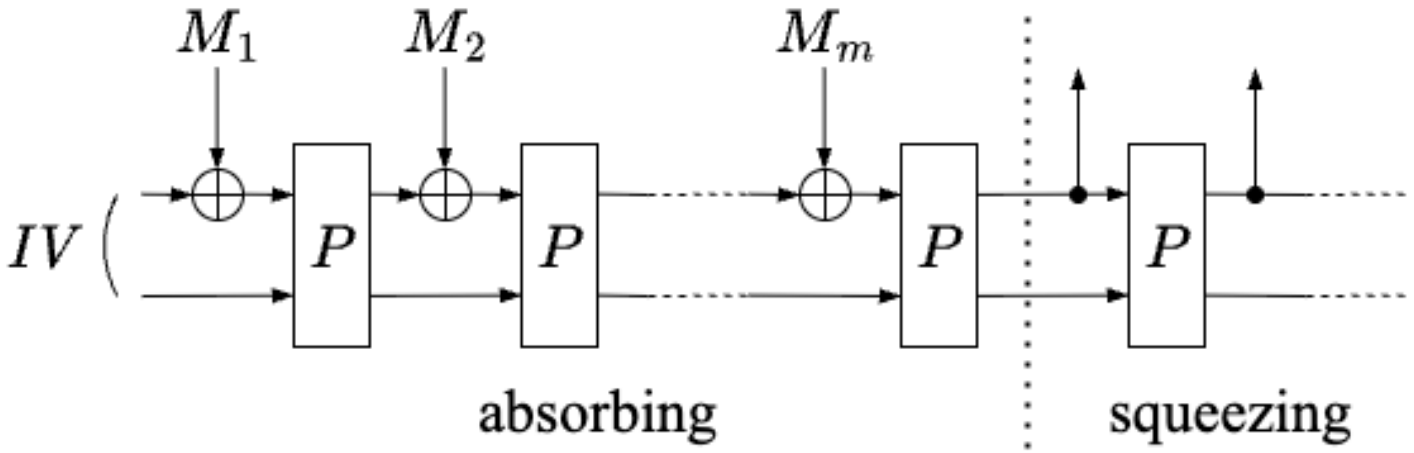
■ Merkle-Damgård構造 [Dam89, Mer89]

代表例：SHA-2 [FIPS180-4]



■ スポンジ構造 [BDPV07]

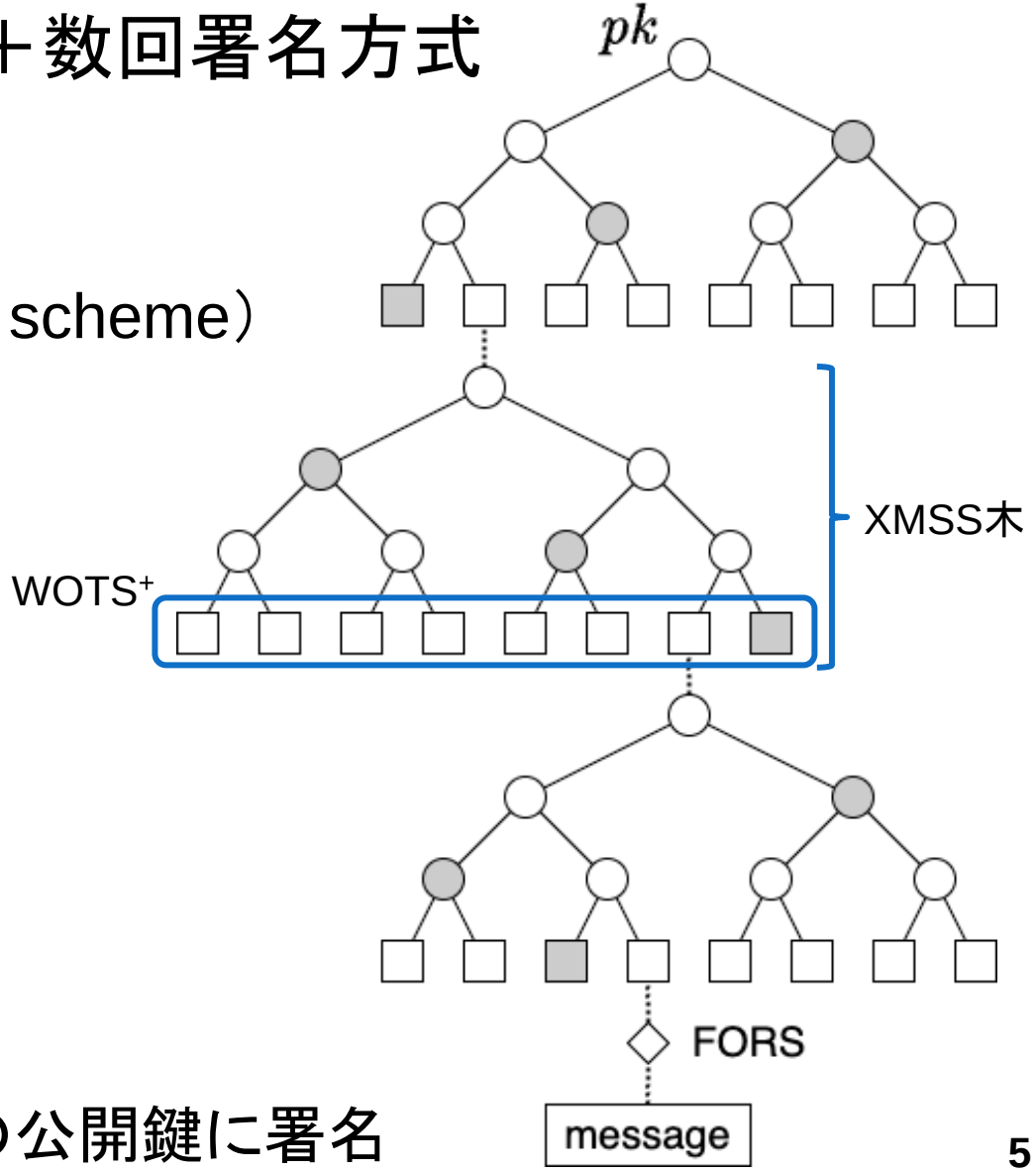
代表例： SHA-3 [FIPS202]



SLH-DSAの概要

■ マークル木を用いた署名方式の階層構造＋数回署名方式

- XMSS（eXtended Merkle Signature Scheme）
  - ➢ マークル木を用いた署名方式
- WOTS$^+$（Winternitz One-Time Signature Plus scheme）
  - ➢ XMSSで用いられる1回署名方式
- Hypertree
  - ➢ XMSS木の階層構造
- FORS（Forest Of Random Subsets）
  - ➢ 数回署名方式

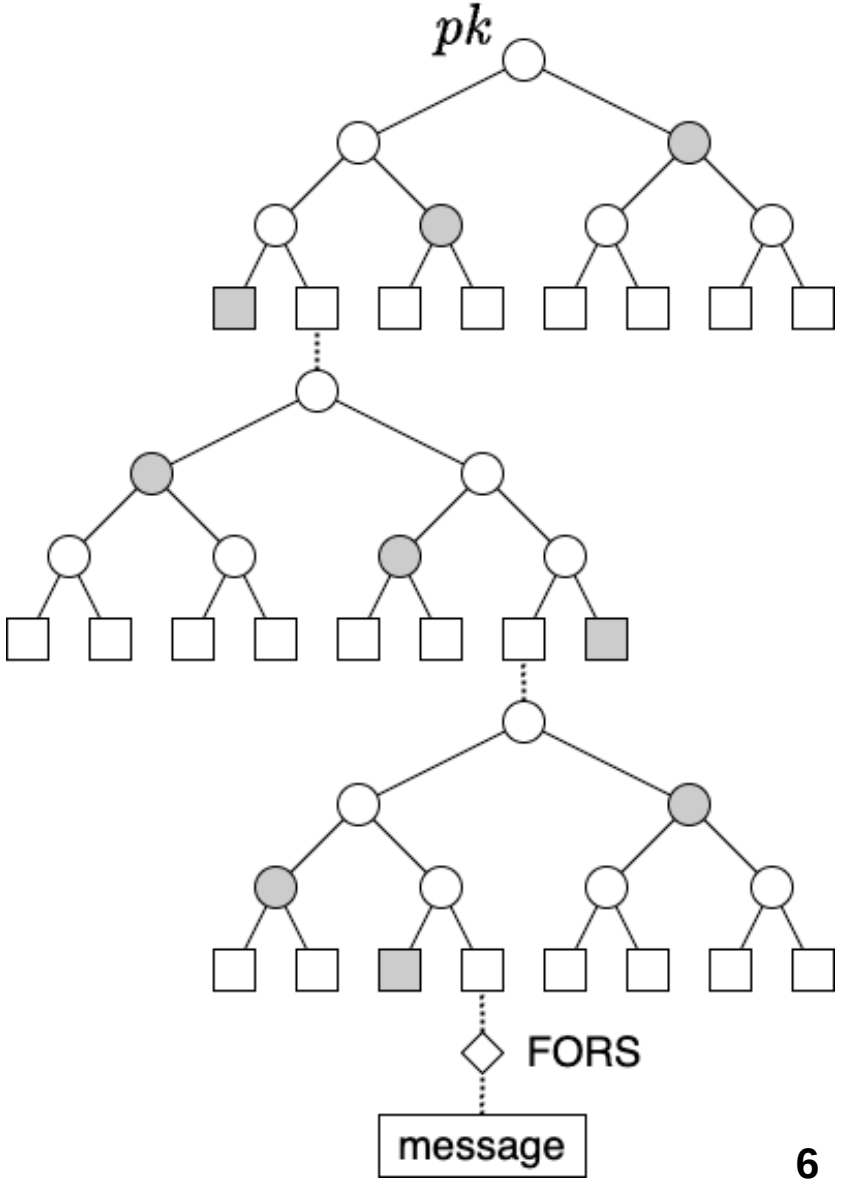


■ 署名の概要

- FORSはメッセージに署名
- 最下層のXMSS木はFORSの公開鍵に署名
- 最上位・中間層のXMSS木は直下のXMSS木の公開鍵に署名

SLH-DSAの公開鍵と秘密鍵

- 公開鍵
  - **PK.** root：hypertreeの最上層のXMSS木の根
  - **PK.** seed：ハッシュ関数への入力の一部
- 秘密鍵
  - **SK.** seed：WOTS$^+$とFORSのすべての秘密鍵の生成に用いられる
  - **SK.** prf：メッセージダイジェストの計算に使用される乱数列の生成に用いられる
- **PK.** root, **PK.** seed, **SK.** seed, **SK.** prfは，すべて長さnバイトの系列



SLH-DSAのハッシュ関数と擬似ランダム関数

- $\mathbf{H}_{msg}(R, \mathbf{PK.seed}, \mathbf{PK.root}, M)$　メッセージダイジェストの計算
- $\mathbf{PRF}_{msg}(\mathbf{SK.prf}, opt_{rand}, M)$　$\mathbf{H}_{msg}$で用いられる乱数Rの計算
- $\mathbf{PRF}(\mathbf{PK.seed}, \mathbf{SK.seed}, \mathbf{ADRS})$　WOTS$^+$, FORSの秘密鍵の計算
- $\mathbf{T}_\ell(\mathbf{PK.seed}, \mathbf{ADRS}, M_\ell)$　WOTS$^+$, XMSS, FORSで用いられる

- 上記4つの関数は以下のいずれかのハッシュ関数を用いて構成される
  - SHA-2（SHA-256, SHA-512）
  - SHA-3（SHAKE256）
- 安全性低下防止のため，各ハッシュ関数に異なる入力が与えられる
  - **PK.seed**は署名者毎に異なる
  - **ADRS**はハッシュ関数が使用される位置を表す

SLH-DSAのパラメータセット

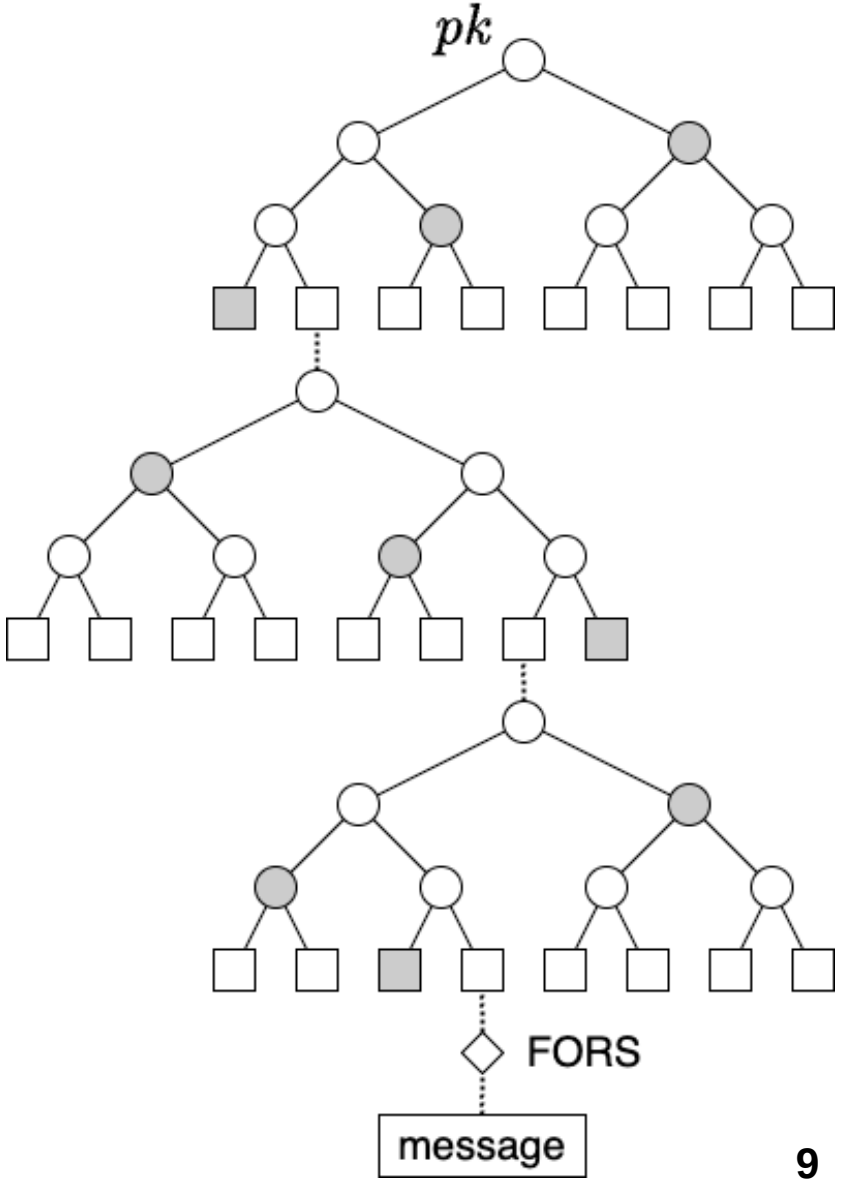
種類	n	h	d	a	k	w	m	安全性	公開鍵長	署名長
128s	16	63	7	12	14	16	30	1	32B	7,856B
128f	16	66	22	6	33	16	34	1	32B	17,088B
192s	24	63	7	14	17	16	39	3	48B	16,224B
192f	24	66	22	8	33	16	42	3	48B	35,664B
256s	32	64	8	14	22	16	47	5	64B	29,792B
256f	32	68	17	9	35	16	49	5	64B	49,856B

- それぞれSHA-2，SHA-3のいずれかを用いて構成（合計12種類）
- sは署名長が短い，fは署名生成が速い
- $\leq 2^{64}$回の署名を可能とする（hはhypertreeの高さ，dはXMSS木の階層数）
- nは$\mathbf{PK}.\mathrm{root}$，$\mathbf{PK}.\mathrm{seed}$，$\mathbf{SK}.\mathrm{seed}$，$\mathbf{SK}.\mathrm{prf}$のバイト長（$\mathbf{PRF}_{msg}$，$\mathbf{PRF}$，$\mathbf{T}_\ell$の出力長）
- mはメッセージダイジェストのバイト長（$\mathbf{H}_{msg}$の出力長）

SLH-DSAの署名アルゴリズム

メッセージMの署名

1. $R = \mathbf{PRF}_{msg}(\mathbf{SK}.\mathrm{prf}, opt_rand, M)$を計算
   - Rは署名者のみが計算できる（秘密鍵$\mathbf{SK}.\mathrm{prf}$が必要）
2. $digest = \mathbf{H}_{msg}(R, \mathbf{PK}.\mathrm{seed}, \mathbf{PK}.\mathrm{root}, M)$を計算
   - $digest$は前半と後半に分割される
3. $digest$の後半により署名に用いる以下を選択
   - 最下層のXMSS木
   - FORSの鍵
4. FORSにより$digest$の前半に署名
5. HypertreeによりFORSの公開鍵に署名
   - Hypertreeの各層でXMSS木が一つずつ構成される



XMSSの構成の基本（マークル木を用いた署名方式）

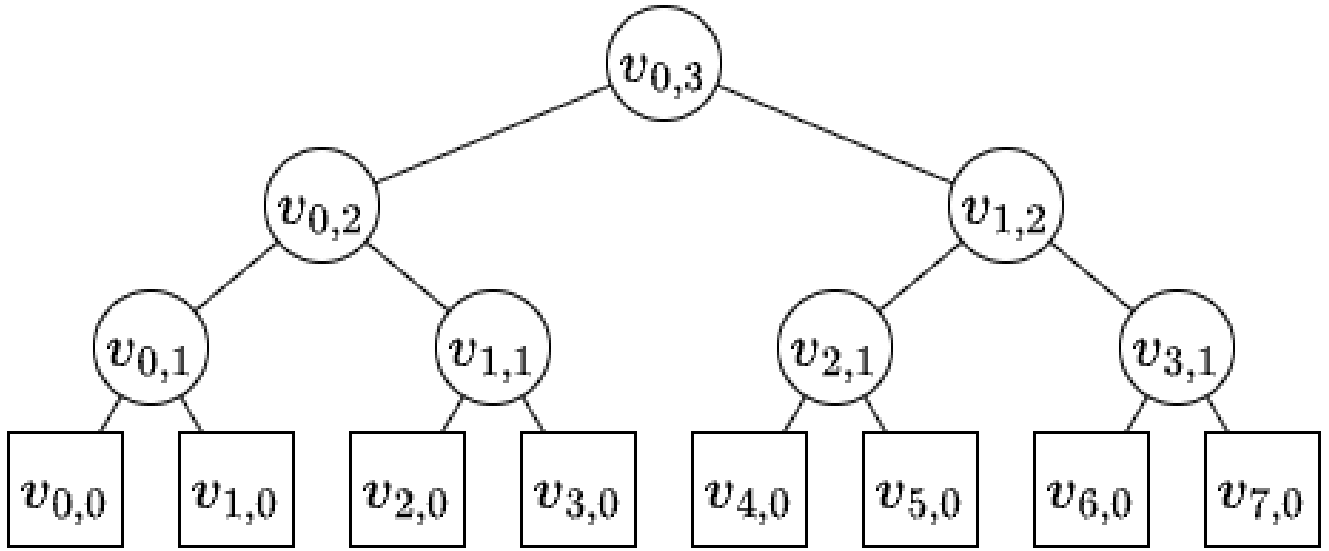
1回署名方式と高さhのマークル木を用いて，2^h回の署名が可能

■ 鍵生成（$h = 3$の例）

1. 1回署名の公開鍵と秘密鍵の組$(pk_0, sk_0), \ldots, (pk_7, sk_7)$を生成
2. 各$i = 0, \ldots, 7$について$H(pk_i)$を葉のラベル$v_{i,0}$とする
3. 各節点のラベルを計算
   - 2つの子節点のラベルのハッシュ値

公開鍵：根のラベル$v_{0,3}$

秘密鍵：$sk_0, \ldots, sk_7$

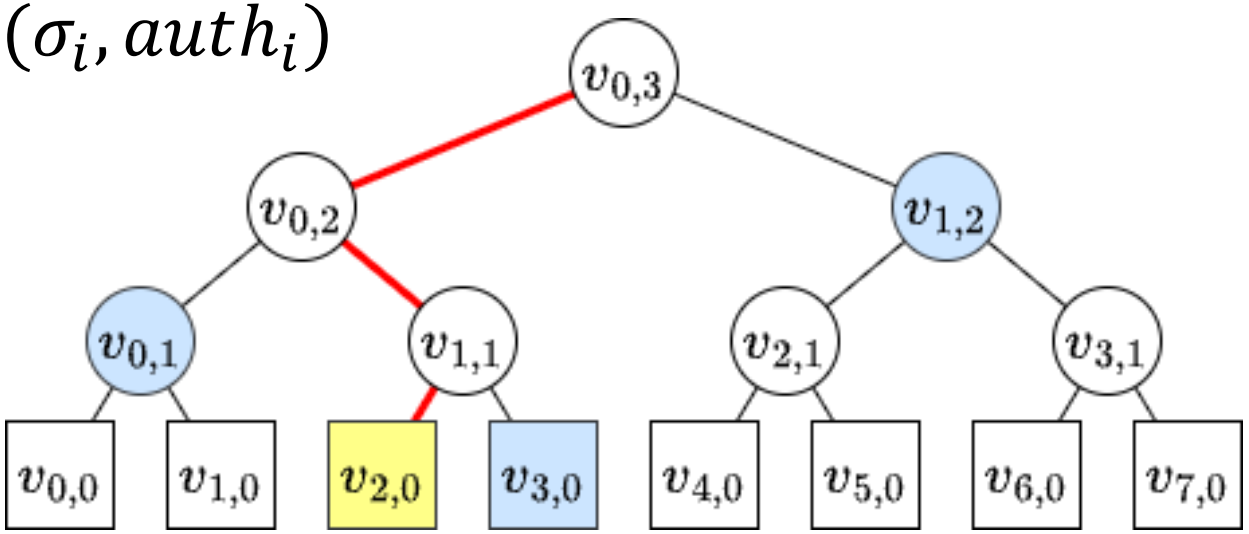


XMSSの構成の基本（マークル木を用いた署名方式）

■ 署名　i番目のデータD_iに対する署名$(\sigma_i, auth_i)$

- σ_iはi番目の秘密鍵sk_iによるD_iの署名
- $auth_i$は$v_{i,0} = H(pk_i)$の認証パス

右図は$i = 2$の例



■ 認証パス

- 各葉のラベルに対して定義される
- 葉のラベルを用いて根のラベルを計算するのに必要な節点のラベルの列
  （葉から根に至る経路上の各節点の経路上にない子節点の列）

■ 検証

1. データD_iと署名$(\sigma_i, auth_i)$から公開鍵の候補$v'_{0,h}$を計算
2. 署名を受理 $\Leftrightarrow v_{0,h} = v'_{0,h}$

WOTS⁺の構成の基本

署名対象のデータ$D \in \{0,1\}^{\log w}$を整数とみなす$(0 \leq D \leq w-1)$

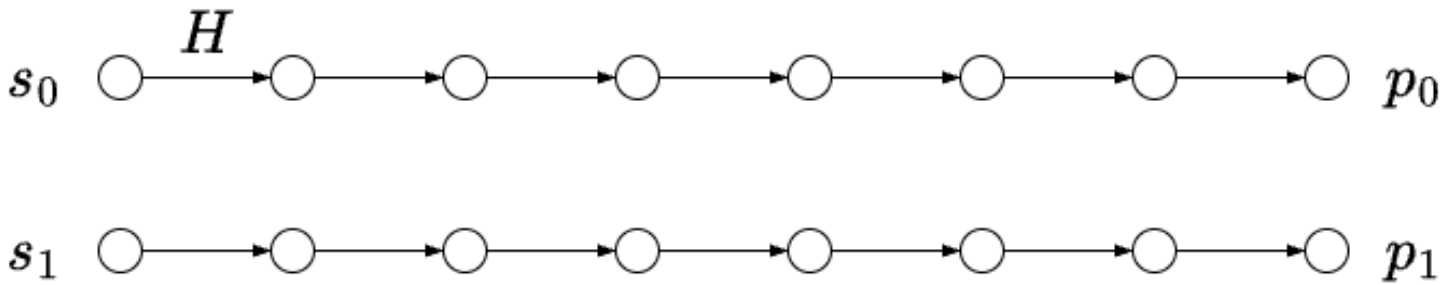
■ 鍵生成

1. 無作為にs_0, s_1を生成
2. ハッシュチェイン$p_i = H^{w-1}(s_i)$を計算

公開鍵: p_0, p_1

秘密鍵: s_0, s_1

■ $w = 8$の例



WOTS⁺の構成の基本

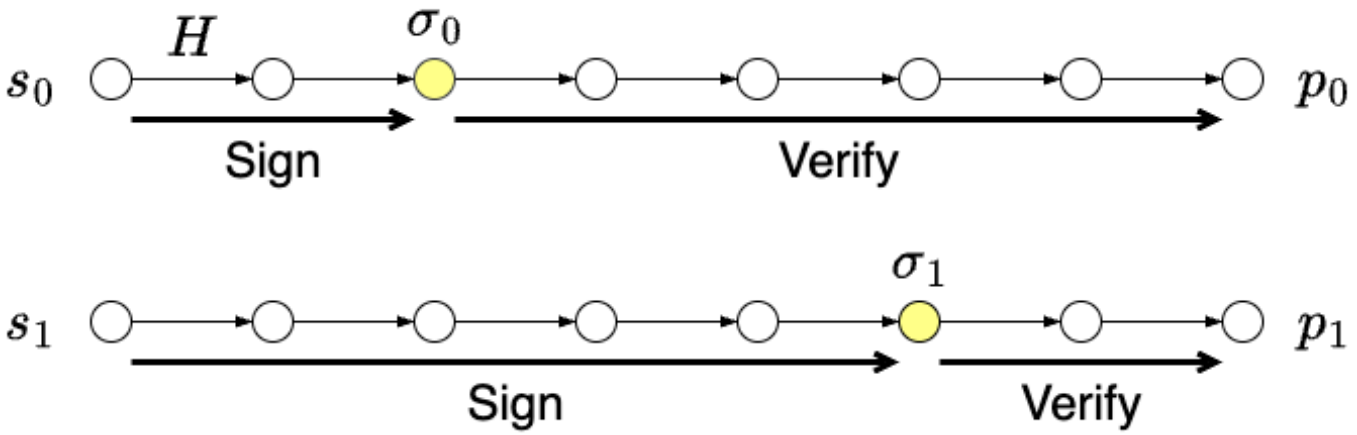
■ 署名　データDに対する署名(σ_0, σ_1)

チェックサム

$$\sigma_0 = H^D(s_0), \quad \sigma_1 = H^{w-1-D}(s_1)$$

■ 検証　$p_0 = H^{w-1-D}(\sigma_0) \wedge p_1 = H^D(\sigma_1) \Leftrightarrow (\sigma_0, \sigma_1)$は$D$の正しい署名

■ $w = 8, D = 2$の例



【注意】チェックサムがないと容易に偽造可能

- 実際は，データを$\log w$ビットごと$D_0, D_1, \ldots$に分割し，$\sum_i (w - 1 - D_i)$とする

FORS（Forest Of Random Subsets）の構成の基本

■ パラメータ　$k, t = 2^a$（k, aは正整数）

■ 鍵生成

$0 \leq i \leq k-1$について

1. 秘密鍵$(sk_{i,0}, \ldots, sk_{i,t-1})$を無作為に選択
2. 高さaのマークル木を構成する
   - $0 \leq j \leq t-1$について，j番目の葉のラベルを$v_{i,j,0} = H(sk_{i,j})$とする

公開鍵：$v_{0,0,a}, \ldots, v_{k-1,0,a}$（$k$個のマークル木の根のラベル）

秘密鍵：$(sk_{0,0}, \ldots, sk_{0,t-1}), \ldots, (sk_{k-1,0}, \ldots, sk_{k-1,t-1})$

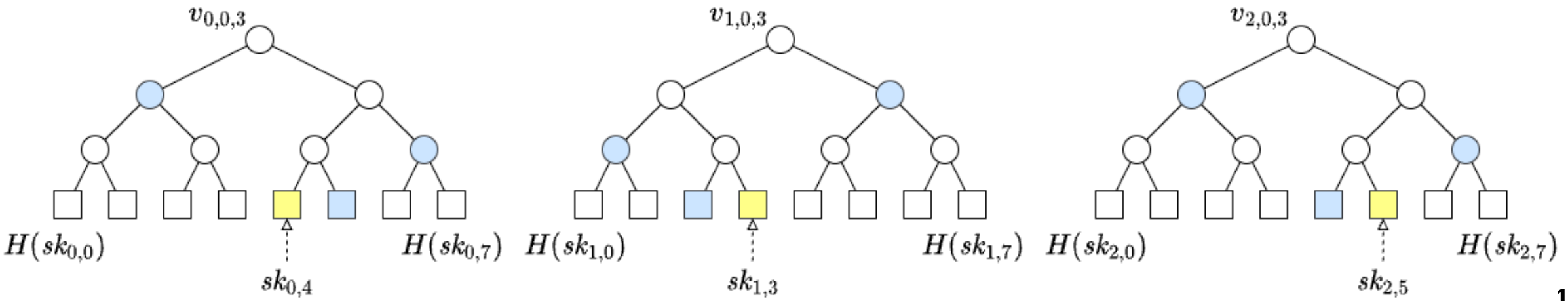
FORS（Forest Of Random Subsets）の構成の基本

■ 署名　データ$D \in \{0,1\}^{ka}$に対する署名

1. データDをk個の長さaビットのブロック$D_0, \ldots, D_{k-1}$に分割
   - 各D_iを2進数とみなすと$0 \leq D_i \leq 2^a - 1 = t - 1$
2. 署名は$(sk_{0,D_0}, \ldots, sk_{k-1,D_{k-1}})$と$(v_{0,D_0,0}, \ldots, v_{k-1,D_{k-1},0})$の認証パス

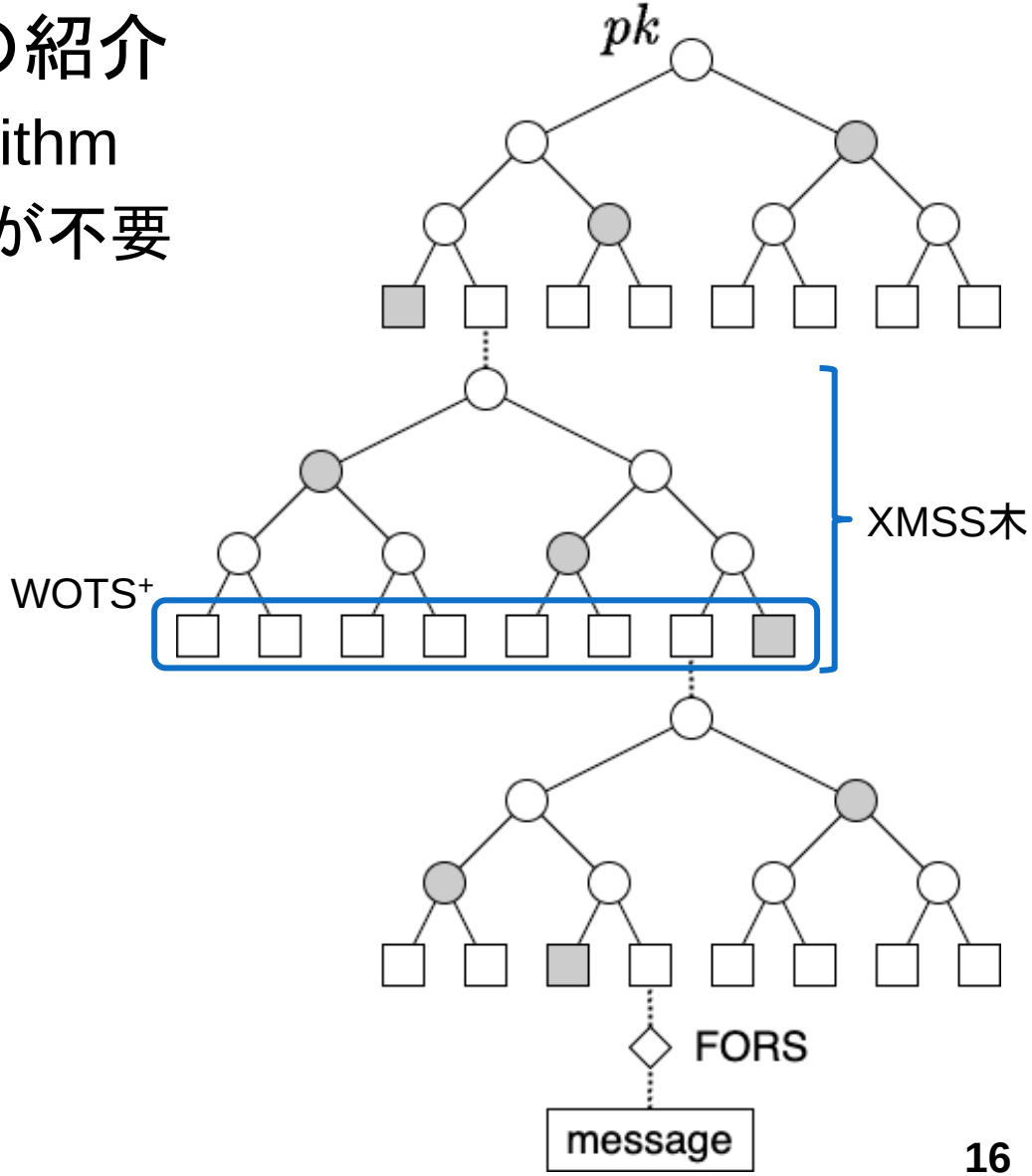
【例】$D = (100,011,101) = (4,3,5)$に対する署名

$(sk_{0,4}, sk_{1,3}, sk_{2,5})$と$(v_{0,4,0}, v_{1,3,0}, v_{2,5,0})$の認証パス



まとめ

- FIPS205のハッシュベース署名SLH-DSAの紹介
  - StateLess Hash-based Digital Signature Algorithm
  - 署名に用いられる秘密鍵の使用・未使用の管理が不要
  - SPHINCS$^+$ ver. 3.1に基づく（ほぼ同じ）
- マークル木を用いた署名方式の階層構造
  ＋数回署名方式
  - XMSS：マークル木を用いた署名方式
  - WOTS$^+$：XMSSで用いられる1回署名方式
  - Hypertree：XMSS木の階層構造
  - FORS：数回署名方式



https://www.cryptrec.go.jp/