Gui

proposed by
Jintai Ding, Ming-Shing Chen, Albrecht Petzoldt, Dieter Schmidt and Bo-Yin Yang

First NIST PQC Standardization Workshop

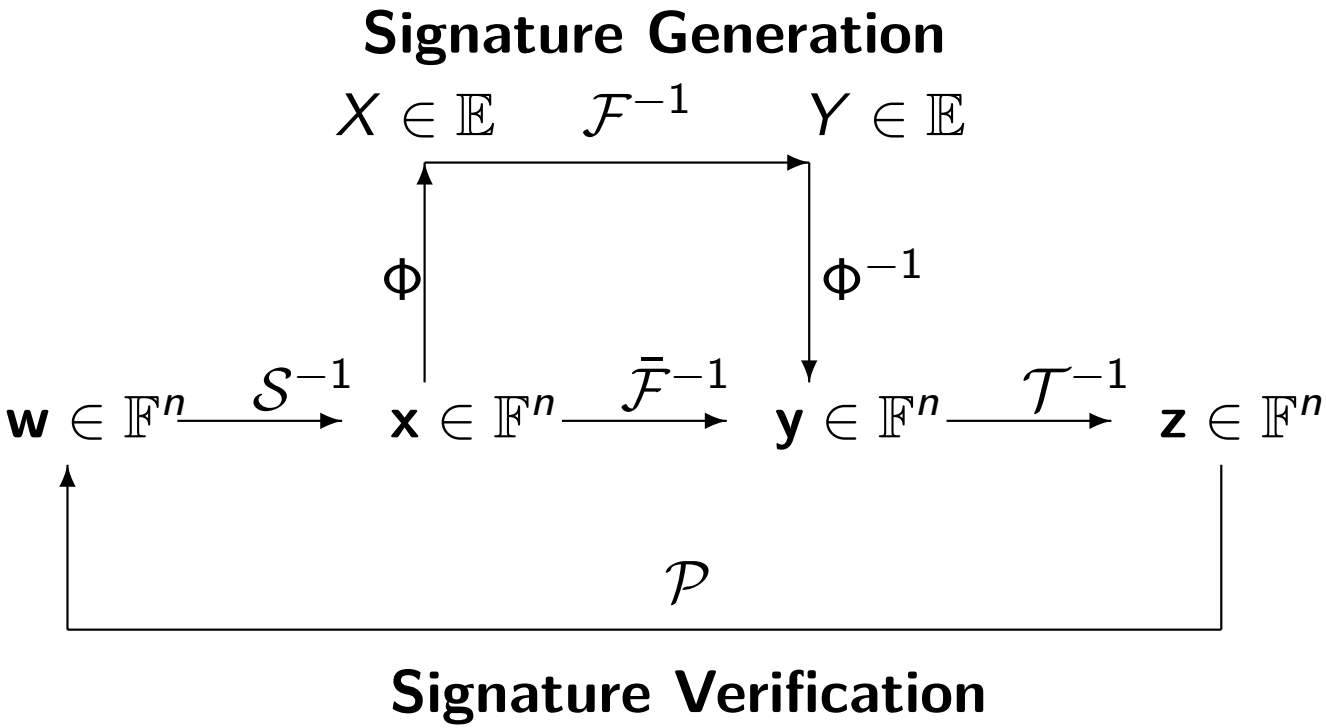
Fort Lauderdale, Florida
04/12/2018

Gui

Type: Signature Scheme

Family: Multivariate Cryptography / Big-Field Schemes

Big Field Signature Schemes



HFEv$^-$ [PatCG 2001] - Key Generation

- BigField + Minus Equations + Vinegar Variation
- central map $\mathcal{F} : \mathbb{F}^v \times \mathbb{E} \to \mathbb{E}$,

$$\mathcal{F}(X) = \sum_{0 \le i \le j}^{q^i + q^j \le D} \alpha_{ij} X^{q^i + q^j} + \sum_{i=0}^{q^i \le D} \beta_i(v_1, \ldots, v_v) \cdot X^{q^i} + \gamma(v_1, \ldots, v_v)$$

  $\Rightarrow \bar{\mathcal{F}} = \Phi^{-1} \circ \mathcal{F} \circ \Phi$ quadratic
- linear maps $\mathcal{S} : \mathbb{F}^n \to \mathbb{F}^{n-a}$ and $\mathcal{T} : \mathbb{F}^{n+v} \to \mathbb{F}^{n+v}$ of maximal rank
- *public key*: $\mathcal{P} = \mathcal{S} \circ \bar{\mathcal{F}} \circ \mathcal{T} : \mathbb{F}^{n+v} \to \mathbb{F}^{n-a}$
- *private key*: $\mathcal{S}$, $\mathcal{F}$, $\mathcal{T}$

Signature Generation

Given: message (hash value) $\mathbf{w} \in \mathbb{F}^{n-a}$

1. Compute $\mathbf{x} = \mathcal{S}^{-1}(\mathbf{w}) \in \mathbb{F}^n$ and $X = \Phi(\mathbf{x}) \in \mathbb{E}$
2. Choose random values for the vinegar variables $v_1, \ldots, v_v$
   Solve $\mathcal{F}_{v_1,\ldots,v_v}(Y) = X$ over $\mathbb{E}$ via Berlekamps algorithm
3. Compute $\mathbf{y} = \Phi^{-1}(Y) \in \mathbb{F}^n$ and $\mathbf{z} = \mathcal{T}^{-1}(\mathbf{y}||v_1|| \ldots ||v_v)$

Signature: $\mathbf{z} \in \mathbb{F}^{n+v}$.

Signature Verification

Given: signature $\mathbf{z} \in \mathbb{F}^{n+v}$, message (hash value) $\mathbf{w} \in \mathbb{F}^{n-a}$

- Compute $\mathbf{w}' = \mathcal{P}(\mathbf{z}) \in \mathbb{F}^{n-a}$
- Accept the signature $\mathbf{z} \Leftrightarrow \mathbf{w}' = \mathbf{w}$.

HFEv- $\rightarrow$ Gui [Asiacrypt 2015]

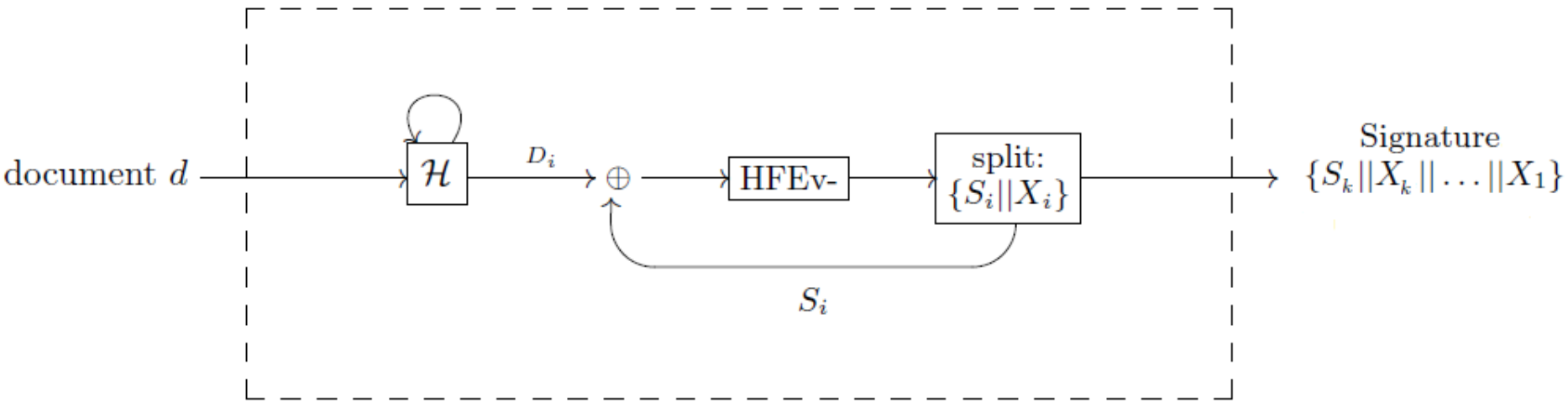
For efficiency reasons, we want to implement HFEv- over GF(2)

Problem: To cover a large hash value, we need many equations in the system $\Rightarrow$ large public key

- QUARTZ $(n, D, a, v) = (103, 129, 3, 4)$ inefficient due to large D, design principles unknown
- new mathematical understanding of the security analysis - degree of regularity - leads to better design with solid security analysis

Signature Generation and Verification

- The HFEv- core is executed k times to prevent birthday attacks
- Single HFEv- signatures are combined to one short signature
- Public equations are evaluated k times while verifying a signature



EUFCMA Security

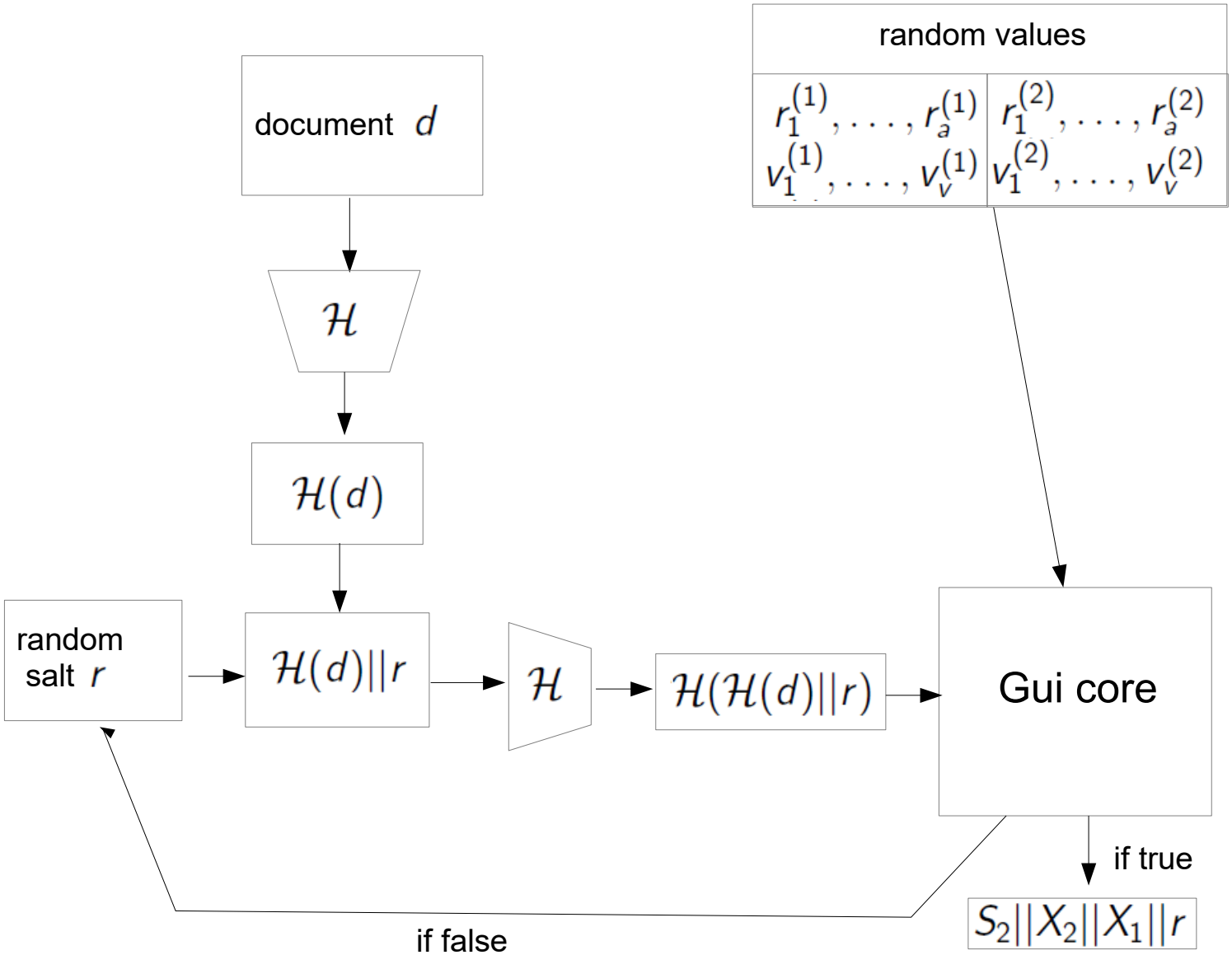
Idea: Use 128 bit seed

Signature Generation:

- For a document d to be signed, compute $\mathcal{H}(d)$ (leads to better performance)
- Choose all random values (Minus and Vinegar variables for all HFEv-layers) at once
- Choose a random salt $r \in \{0,1\}^{128}$ and check if Gui produces a signature $S_k, X_k \ldots, X_1$ for $\mathcal{H}(\mathcal{H}(d)||r)$; if yes, output $S_k||X_k|| \ldots ||X_1||r$, otherwise choose new random salt

Signature Verification: Check, if $S_k||X_k|| \ldots ||X_1||r$ is a valid signature of $\mathcal{H}(\mathcal{H}(d)||r)$.

EUFCMA-Secure Signature Generation (for $k = 2$)



Design Decisions

- Choose the repetition factor k to be 2
  - sufficient for security
  - $k \geq 3$ will increase signatures and slow down the scheme
- Choose a and v to be as equal as possible
  but: $(n - a)$ must be a multiple of 8 (for efficiency reasons)

Implementation

Field Representations

- Build GF(256) as a "Tower-Field", i.e.

$$\begin{aligned} \mathbb{F}_4 &= \mathbb{F}_2[e_1]/(e_1^2 + e_1 + 1), \\ \mathbb{F}_{16} &= \mathbb{F}_4[e_2]/(e_2^2 + e_2 + 1), \\ \mathbb{F}_{256} &= \mathbb{F}_{16}[e_3]/(e_3^2 + e_3 + e_1 e_2). \end{aligned}$$

- Build the finite fields used by Gui as extension fields of GF(256)

$$\begin{aligned} \mathbb{F}_{2^{184}} &= \mathbb{F}_{256}[X]/(X^{23} + X^3 + X + 0x2), \\ \mathbb{F}_{2^{312}} &= \mathbb{F}_{256}[X]/(X^{39} + X^2 + X + 0x2), \\ \mathbb{F}_{2^{448}} &= \mathbb{F}_{256}[X]/(X^{56} + 0x2 \cdot X^3 + X + 0x10). \end{aligned}$$

Central Step: Inverting the HFEv- central map

Compute $\gcd(\mathcal{F}_V(Y) - X, Y^{2^n} - Y)$ or $Y^{2^n} \bmod \mathcal{F}_V(Y)$.

- Recursively raise the lower degree polynomial Y^{2^m} to the power of 2.

$$\begin{aligned} & (Y^{2^m} \bmod \mathcal{F}_V(Y))^2 \bmod \mathcal{F}_V(Y) \\ = & (\sum_{i<2^m} b_i Y^i)^2 \bmod \mathcal{F}_V(Y) \\ = & (\sum_{i<2^m} b_i^2 Y^{2i}) \bmod \mathcal{F}_V(Y) \end{aligned}$$

- Starting relation: $\mathcal{F}_V(Y) = Y^D + \sum_{0\le i\le j, 2^i+2^j<D} a_{ij} Y^{2^i+2^j}$
- Prepare a table for $Y^{2i} \bmod \mathcal{F}_V(Y)$ first.
  For large D ($\mathcal{F}_V(Y)$ sparse) we can do the reduction by linear folding.
- Square all the coefficients b_i in $Y^{2^m} \bmod \mathcal{F}_V(Y)$ and multiply them to the Y^{2i}s in the table.

optimized implementation: use PCLMULQDQ instructions to speed up the computations

$\Rightarrow$ Much more details in the proposal

Security

- No security proof / reduction to hard problem
- Security is measured by analyzing the complexity of known attacks
  - direct attacks
  - Rank attacks
  - distinguishing based attack [PQ Crypto 2018]
- A detailed analysis of these attacks can be found in the proposal

Proposed Parameters

	parameters (n, D, a, v, k)	public key size (kB)	private key size (kB)	signature size (bit)	NIST security category
Gui-184	(184, 33, 16, 16, 2)	416.3	19.1	360	I, II
Gui-312	(312, 129, 24, 20, 2)	1,955.1	59.3	504	III, IV
Gui-448	(448, 513, 32, 28, 2)	5,789.2	155.9	664	V,VI

Signature size includes 128-bit seed

Performance

scheme		key generation	signature generation	signature verification
Gui-184	cycles	2,408M / 704M	1,910M / 34M	152k / 169k
	time (ms)	730 / 213	579 / 10.4	0.046 / 0.051
	memory (MB)	3.5 MB / 3.5 MB	3.4 MB / 3.4 MB	3.3 MB / 3.3 MB
Gui-312	cycles	43,817M / 4,790M	25,436M / 1,757M	/595k
	time (ms)	13,227 / 1,452	7,707 / 532	0.256 / 0.181
	memory (MB)	5.4MB / 5.4 MB	3.8 MB / 3.6 MB	5.0 MB / 5.0 MB
Gui-448	cycles	239,502M / 32,247M	872,949M / 86,086M	1,787k/ 3,385k
	time (ms)	72,585 / 9,772	264,530 / 26,086	0.542 / 1.025
	memory (MB)	17.7 MB / 9.2 MB	10.7 MB / 10.7 MB	8.7 MB / 8.7 MB

Performance on
NIST Reference Platform (Intel Xeon E3-1225 v5 (Skylake), 3.3 GHz, 64 GB RAM, no special processor instructions) /
Intel Xeon E3-1225 v5 (Skylake), 3.3 GHz, 64 GB RAM, Use of PCLMULQDQ instructions

$\Rightarrow$ By using PCLMULQDQ instructions, we can achieve a speed up of key / signature generation by about 90 %.

Advantages and Limitations

Advantages

- Security well understood
- very short signatures
- time constant implementation

Limitations

- Large key sizes
- Rather slow (especially for high levels of security)