WAVE

A Code-based Hash and Sign Signature Scheme

Gustavo Banegas, Kévin Carrier, André Chailloux, Alain Couvreur, **Thomas Debris-Alazard**, Philippe Gaborit, Pierre Karpman, Johanna Loyer, Ruben Niederhagen, Nicolas Sendrier, Benjamin Smith and Jean-Pierre Tillich

Inria, École Polytechnique

ROADMAP

1. Wave: standardization candidate (NIST),
2. Next steps,
3. Code-based hash and sign,
4. Design Rationale: Wave Trapdoor,
5. Leakage free signatures,
6. Removing Approximation in Prange.

https://wave-sign.org

WAVE: STANDARDIZATION CANDIDATE (NIST)

GPV FRAMEWORK

Wave is a **hash and sign** digital signature scheme.

By proving that signatures are leakage-free,

$\longrightarrow$ Wave instantiates Gentry-Peikert-Vaikuntanathan (**GPV**) framework
like Falcon, Squirrels, HuFu

But Wave security relies on **coding** problems

ADVANTAGES

Even if parameters are highly conservative

- **Short signatures**: linear scaling in the security

Post-quantum target security	Level I	Level III	Level V
Signature length (**Bytes**)	822	1249	1644

- **Fast Verification**: (Intel Core i5-1135G7 platform at 2.40GHz)

Post-quantum target security	Level I	Level III	Level V
Verification (MCycles)	1.2	2.5	4.3

- Immune to statistical attacks.
- Proven secure (Q)ROM with tight reductions.

LIMITATIONS

- Big public-key: quadratic scaling in the security

Post-quantum target security	Level I	Level III	Level V
Public-key size (MBytes)	3.6	7.8	13.6

- Signing and key generation rely on Gaussian elimination on large matrices
- Security based on fairly new assumption (2018): distinguishing random and generalized $(U \mid U+V)$-codes

NEXT STEPS

ABOUT PARAMETERS

Wave parameters are highly conservative!

Attack model:

Cost of $\mathcal{A}$ to solve $\mathcal{P}$:

$$\alpha \stackrel{\text{def}}{=} \lim_{n \to +\infty} \frac{1}{n} \log_2 \text{Time}\left(\mathcal{A}\right)$$

Then choose n s.t:

$$\alpha n = \lambda \qquad (\alpha \approx 0.0149)$$

$\longrightarrow$ It ignores (super-)polynomial factors and memory access!

For instance: considered attack to forge a signature

$$\text{Time} = P(\lambda)2^{\lambda} \quad \text{and} \quad \text{Memory} = Q(\lambda)2^{\lambda}.$$

Next Step:

Providing parameters for "concrete" security.

A MORE OPTIMIZED/SECURE IMPLEMENTATION

Wave reference implementation

- portable *C99*,
- KeyGen and Sign in constant-time,
- bit-sliced arithmetic over $\mathbb{F}_3$.

Bottleneck of Wave: **Gaussian elimination** on big matrices/**memory access**

(it impacts key generation and signing not verification)

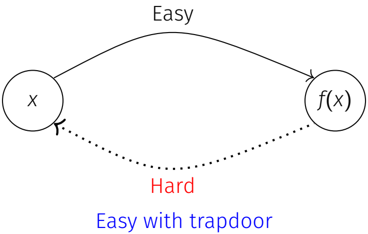
Next Step:

- Providing **optimized** implementation: AVX,
  $\longrightarrow$ Wavelet: AVX2 (intel) & ARM CORTEX M4 **in verification** (2x faster),
- Providing a Wave version with **countermeasures, maskings**,
- Providing (friendly) tools to ensure that Wave is properly implemented.

CODE-BASED HASH AND SIGN

FULL DOMAIN HASH SIGNATURE SCHEME

- $\texttt{Hash}(\cdot)$ hash function,
- f trapdoor one-way function



- To sign $\mathbf{m}$:

$$\text{Compute } \sigma \in f^{-1}(\texttt{Hash}(\mathbf{m})).$$

> f needs to be surjective!

- To verify $(\mathbf{m}, \sigma)$:

$$\text{Check } f(\sigma) \stackrel{?}{=} \texttt{Hash}(\mathbf{m}).$$

CODE-BASED ONE-WAY FUNCTION (1)

$\longrightarrow$ Coding theory provides one-way functions!

- A **$[n, k]$-code** $\mathcal{C}$ is a defined as a k dimension subspace of $\mathbb{F}_q^n$.
- $\mathbb{F}_q^n$ embedded with **Hamming weight**,

$$\forall \mathbf{x} \in \mathbb{F}_q^n, \qquad |\mathbf{x}| \stackrel{\text{def}}{=} \sharp \left\{ i,\ \mathbf{x}(i) \neq 0 \right\}.$$

CODE-BASED ONE-WAY FUNCTION (2)

One-way in code-based crypto:

$$f_w : (\mathbf{c}, \mathbf{e}) \in \mathcal{C} \times \{\mathbf{e} : |\mathbf{e}| = w\} \longmapsto \mathbf{c} + \mathbf{e}.$$

(inverting f_w: decoding $\mathcal{C}$ at distance w)

$\longrightarrow$ To hope f_w surjective: choose noise distance w large enough

CODE-BASED ONE-WAY FUNCTION (2)

One-way in code-based crypto:

$$f_w : (\mathbf{c}, \mathbf{e}) \in \mathcal{C} \times \{\mathbf{e} : |\mathbf{e}| = w\} \longmapsto \mathbf{c} + \mathbf{e}.$$

(inverting f_w: decoding $\mathcal{C}$ at distance w)

$\longrightarrow$ To hope f_w surjective: choose noise distance w large enough

But, be careful...

w parametrizes the hardness of inverting f_w!

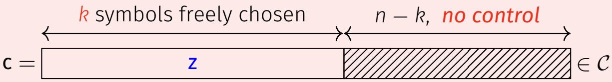
$\longrightarrow$ for some w, it is easy to invert f_w...

Inverting f_w:

- Given: $[n, k]$-$\mathcal{C}$, $\mathbf{y}$ **uniformly distributed** over $\mathbb{F}_q^n$ and w,
- Find: $\mathbf{c} \in \mathcal{C}$ such that $|\mathbf{y} - \mathbf{c}| = w$.

Fact: by linear algebra (Gaussian elimination)

$\mathcal{C}$ has dimension k: $\forall \mathbf{z} \in \mathbb{F}_q^k$, easy to compute $\mathbf{c} \in \mathcal{C}$ such that,



Inverting f_w:

- Given: $[n, k]$-$\mathcal{C}$, $\mathbf{y}$ **uniformly distributed** over $\mathbb{F}_q^n$ and w,
- Find: $\mathbf{c} \in \mathcal{C}$ such that $|\mathbf{y} - \mathbf{c}| = w$.

Fact: by linear algebra (Gaussian elimination)

$\mathcal{C}$ has dimension k: $\forall \mathbf{z} \in \mathbb{F}_q^k$, easy to compute $\mathbf{c} \in \mathcal{C}$ such that,

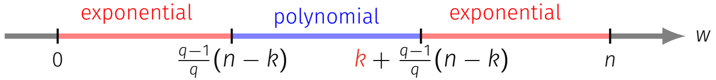
k symbols freely chosen | $n-k$, *no control*

$\mathbf{c} = \big[\ \mathbf{z}\ \big|\ ///////\ \big] \in \mathcal{C}$

Given a uniform $\mathbf{y} \in \mathbb{F}_q^n$: compute $\mathbf{c} \in \mathcal{C}$,

$\mathbf{y} - \mathbf{c} = \big[\ \mathbf{e}\ \big|\ \mathbf{e}''\ \big]$

k symbols (**to choose, guide the distance**) | $n-k$ symbols (Unif Distributed over $\mathbb{F}_q$)



- Public data: a hash function $\texttt{Hash}(\cdot)$, an $[n,k]$-code $\mathcal{C}$ and,

$$w \notin \left[\frac{q-1}{q}(n-k), k + \frac{q-1}{q}(n-k)\right] \qquad (\textit{signing distance})$$

- Signing $\mathbf{m}$:

1. Hashing: $\mathbf{m} \longrightarrow \mathbf{y} \stackrel{\text{def}}{=} \texttt{Hash}(\mathbf{m}) \in \mathbb{F}_q^n$,
2. Decoding: find **with a trapdoor** $\mathbf{c} \in \mathcal{C}$ such that $|\mathbf{y} - \mathbf{c}| = w$.

- Verifying $(\mathbf{m}, \mathbf{c})$:

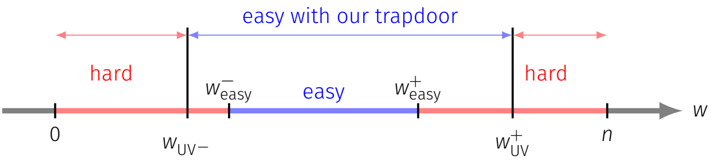
$$\mathbf{c} \in \mathcal{C} \quad \text{and} \quad |\texttt{Hash}(\mathbf{m}) - \mathbf{c}| = w.$$

Security:

Signing distance w s.t hard to find $\mathbf{c} \in \mathcal{C}$ at distance w

$\longrightarrow$ Unless to own a secret/trapdoor structure on $\mathcal{C}$!

DECODING WITH OUR TRAPDOOR

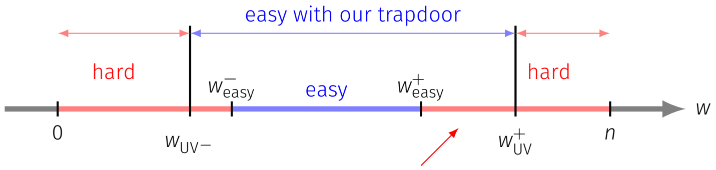


Trapdoor:

An $[n, k]$-code $\mathcal{C}$ with a peculiar structure enabling to decode at distance $w \notin [w_{\text{easy}}^-, w_{\text{easy}}^+]$

Security:

$\mathcal{C}$ indistinguishable from a random code (unless to know its peculiar structure)



Trapdoor:

An $[n, k]$-code $\mathcal{C}$ with a peculiar structure enabling to decode at distance $w \notin [w_{\text{easy}}^{-}, w_{\text{easy}}^{+}]$

Security:

$\mathcal{C}$ indistinguishable from a random code (unless to know its peculiar structure)

DESIGN RATIONALE: WAVE TRAPDOOR

SOME NOTATION

- Vector permutation:

$$\mathbf{x} = (\mathbf{x}(i))_{1 \le i \le n} \in \mathbb{F}_q^n \;\; ; \;\; \pi \text{ permutation of } \{1, \dots, n\}.$$

$$\mathbf{x}^{\pi} \stackrel{\text{def}}{=} (\mathbf{x}(\pi(i)))_{1 \le i \le n}$$

- Component-wise product:

$$\mathbf{a} \star \mathbf{x} \stackrel{\text{def}}{=} (\mathbf{a}(i)\mathbf{x}(i))_{1 \le i \le n}$$

TRAPDOOR: GENERALIZED (U | U+V)-CODES

Generalized $(U \mid U+V)$-codes:

Let U and V be $[n/2, k_U]$ and $[n/2, k_V]$-codes

$$\mathcal{C} \stackrel{\text{def}}{=} \left\{ (\mathbf{x}_U + \mathbf{b} \star \mathbf{x}_V \mid \mathbf{c} \star \mathbf{x}_U + \mathbf{d} \star \mathbf{x}_V)^{\pi} \ : \ \mathbf{x}_U \in U \text{ and } \mathbf{x}_V \in V \right\}$$

where π permutation, $\mathbf{b}, \mathbf{c}, \mathbf{d} \in \mathbb{F}_q^{n/2}$ verify $\mathbf{c}(i) \neq 0$ and $\mathbf{d}(i) - \mathbf{b}(i)\mathbf{c}(i) = 1$.

$\longrightarrow$ It defines a code with dimension $k \stackrel{\text{def}}{=} k_U + k_V$

Secret-key/Trapdoor: $U, V, \mathbf{b}, \mathbf{c}, \mathbf{d}$ and π.

Security assumption: Distinguishing Wave Key (DWK)

Hard to distinguish random and generalized $(U \mid U+V)$ codes.

OUR DECODING ALGORITHM (1)

Secret-key/Trapdoor: $U, V, \mathbf{b}, \mathbf{c}, \mathbf{d}$ and π.

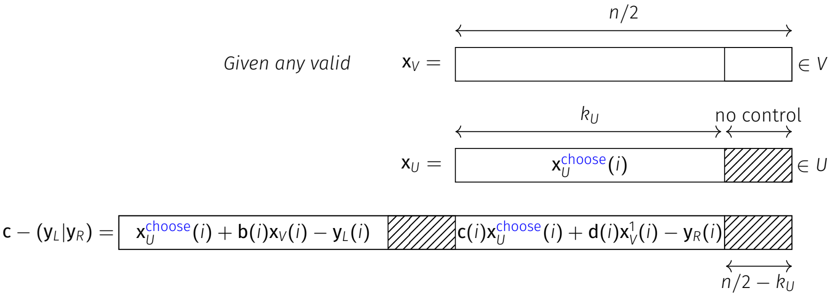
1. Given $\texttt{Hash}(\mathbf{m}) = \mathbf{y} \in \mathbb{F}_q^n$: decompose $\mathbf{y} = (\mathbf{y}_L \mid \mathbf{y}_R)^{\pi}$,
2. Compute any $\mathbf{x}_V \in V$ with Prange Algorithm,
3. Using Prange Algorithm: compute $\mathbf{x}_U \in U$ by **choosing k_U symbols $\mathbf{x}_U(i)$'s** such that

$$\begin{cases} \mathbf{x}_U(i) + \mathbf{b}(i)\mathbf{x}_V(i) \neq \mathbf{y}_L(i) \\ \mathbf{c}(i)\mathbf{x}_U + \mathbf{d}(i)\mathbf{x}_V(i) \neq \mathbf{y}_R(i) \end{cases}$$

(i) $q \geq 3$, (ii) $\mathbf{c}(i) \neq 0$ and (iii) $\mathbf{d}(i) - \mathbf{b}(i)\mathbf{c}(i) = 1$.

4. Return $\mathbf{c} \stackrel{\text{def}}{=} (\mathbf{x}_U + \mathbf{b} \star \mathbf{x}_V \mid \mathbf{c} \star \mathbf{x}_U + \mathbf{d} \star \mathbf{x}_V)^{\pi} \in \mathcal{C}$ (public code).

What is the (typical) distance w between $\mathbf{y}$ and $\mathbf{c}$?



- Choose k_U symbols $\mathbf{x}_U^{\text{choose}}(i)$ such that: $\begin{cases} \mathbf{x}_U^{\text{choose}}(i) + \mathbf{b}(i)\mathbf{x}_V(i) - \mathbf{y}_L(i) \neq 0 \\ \mathbf{c}(i)\mathbf{x}_U^{\text{choose}}(i) + \mathbf{d}(i)\mathbf{x}_V(i) - \mathbf{y}_R(i) \neq 0 \end{cases}$

Typical distance:

$$w = 2k_U + 2\frac{q-1}{q}(n/2 - k_U) > w_{\text{easy}}^{+} = (k_U + k_V) + \frac{q-1}{q}(n - (k_U + k_V))$$

as soon as: $k_U > k_V$ (parameter constraint in Wave)

BE CAREFUL: A HUGE ISSUE

Collecting signatures:

$$(\mathsf{x}_U + \mathsf{b} \star \mathsf{x}_V \mid \mathsf{c} \star \mathsf{x}_U + \mathsf{d} \star \mathsf{x}_V)^{\pi}$$

may enable to recover the secret, for instance π...

BE CAREFUL: A HUGE ISSUE

Collecting signatures:

$$(\mathbf{x}_U + \mathbf{b} \star \mathbf{x}_V \mid \mathbf{c} \star \mathbf{x}_U + \mathbf{d} \star \mathbf{x}_V)^{\pi}$$

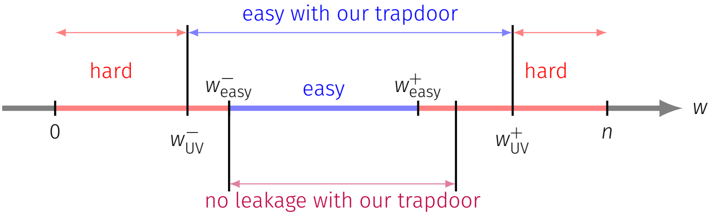
may enable to recover the secret, for instance π...

Above procedure **leaks quickly** π...

Proper Wave **specification/implementation:**

Choose carefully internal distribution and perform rejection sampling

to produce signatures immune to statistical attacks



AN IMPORTANT CHOICE OF PARAMETERS

In what follows:

We will work in $\mathbb{F}_3$, $q = 3$.

LEAKAGE FREE SIGNATURES

A signature: $x \in f^{-1}(y)$.

$\longrightarrow$ x computed via a trapdoor/secret!

Ideal situation:

x distribution independent of the secret

$\longrightarrow$ For instance: x uniform over its domain when y uniform

A hard problem

In our case: exponential number of preimages

OUR AIM

Given uniform $\mathbf{y}$: compute $(\mathbf{x}_U + \mathbf{b} \star \mathbf{x}_V \mid \mathbf{c} \star \mathbf{x}_U + \mathbf{d} \star \mathbf{x}_V)^{\pi}$ such that

$\mathbf{e}^{\text{sgn}} \stackrel{\text{def}}{=} \mathbf{y} - (\mathbf{x}_U + \mathbf{b} \star \mathbf{x}_V \mid \mathbf{c} \star \mathbf{x}_U + \mathbf{d} \star \mathbf{x}_V)^{\pi}$ uniform over words of Hamming weight w.

Given uniform $\mathbf{y}$: compute $(\mathbf{x}_U + \mathbf{b} \star \mathbf{x}_V \mid \mathbf{c} \star \mathbf{x}_U + \mathbf{d} \star \mathbf{x}_V)^{\pi}$ such that

$\mathbf{e}^{\mathrm{sgn}} \stackrel{\text{def}}{=} \mathbf{y} - (\mathbf{x}_U + \mathbf{b} \star \mathbf{x}_V \mid \mathbf{c} \star \mathbf{x}_U + \mathbf{d} \star \mathbf{x}_V)^{\pi}$ uniform over words of Hamming weight w.

Important fact: as $\mathbf{d}(i) - \mathbf{b}(i)\mathbf{c}(i) = 1$ for all i,

$$\varphi : (\mathbf{z}_U, \mathbf{z}_V) \longmapsto (\mathbf{z}_U + \mathbf{b} \star \mathbf{z}_V \mid \mathbf{c} \star \mathbf{z}_U + \mathbf{d} \star \mathbf{z}_V)^{\pi} \text{ } \textit{bijection}.$$

1. Write $\mathbf{y} = (\mathbf{y}_U + \mathbf{b} \star \mathbf{y}_V \mid \mathbf{c} \star \mathbf{y}_U + \mathbf{d} \star \mathbf{y}_V)^{\pi}$
2. Deduce that $\mathbf{e}^{\mathrm{sgn}} = (\mathbf{e}_U + \mathbf{b} \star \mathbf{e}_V \mid \mathbf{c} \star \mathbf{e}_U + \mathbf{d} \star \mathbf{e}_V)^{\pi}$ where $\begin{cases} \mathbf{e}_V \stackrel{\text{def}}{=} \mathbf{y}_V - \mathbf{x}_V \\ \mathbf{e}_U \stackrel{\text{def}}{=} \mathbf{y}_U - \mathbf{x}_U \end{cases}$

Here $\mathbf{x}_V$ and $\mathbf{x}_U$ are computed via Prange algorithm...

$$\mathbf{e}^{\text{sgn}} \stackrel{\text{def}}{=} (\mathbf{e}_U + \mathbf{b} \star \mathbf{e}_V \mid \mathbf{c} \star \mathbf{e}_U + \mathbf{d} \star \mathbf{e}_V)^{\pi} \quad \text{and} \quad \mathbf{e}^{\text{unif}} \text{ unif word of weight } w.$$

$$\longrightarrow \text{Write: } \mathbf{e}^{\text{unif}} = (\mathbf{e}_U^{\text{unif}} + \mathbf{b} \star \mathbf{e}_V^{\text{unif}} \mid \mathbf{c} \star \mathbf{e}_U^{\text{unif}} + \mathbf{d} \star \mathbf{e}_V^{\text{unif}})^{\pi}$$

We would like,

$$\mathbf{e}^{\text{sgn}} \sim \mathbf{e}^{\text{unif}}$$

In a first step we want,

$$\mathbf{e}_V \sim \mathbf{e}_V^{\text{unif}} \quad \text{where} \quad \mathbf{e}_V = \mathbf{y}_V - \mathbf{x}_V = \mathbf{y}_V - \text{Prange}(V, \mathbf{y}_V)$$

Important remark (function of weight):

$$\mathbb{P}\left(\mathbf{e}_V^{\text{unif}} = \mathbf{x}\right) = \frac{1}{\sharp\{\mathbf{y} : |\mathbf{y}| = t\}} \mathbb{P}\left(\left|\mathbf{e}_V^{\text{unif}}\right| = t\right) \quad \text{when } |\mathbf{x}| = t.$$

Approximation: Distribution of Prange algorithm, only function of the weight

$$\mathbb{P}(\text{Prange}(\cdot) = \mathbf{x} \mid |\text{Prange}(\cdot)| = t) = \frac{1}{\sharp\{\mathbf{y} : |\mathbf{y}| = t\}} \quad \text{when } |\mathbf{x}| = t.$$

$\longrightarrow$ Uniformity property: **enough to reach** $|\mathbf{e}_V| \sim |\mathbf{e}_V^{\text{unif}}|$ as distribution

GUIDE THE WEIGHT OF $\mathbf{e}_V$

- We first look for $\mathbb{E}(|\mathbf{e}_V|) = \mathbb{E}(|\mathbf{e}_V^{\text{unif}}|)$

$$\mathbf{e}_V = \underbrace{\boxed{\quad \mathbf{e}'_V \quad}}_{k_V \text{ symbols}}\underbrace{\boxed{\quad \mathbf{e}''_V \quad}}_{n/2 - k_V \text{ symbols}}$$

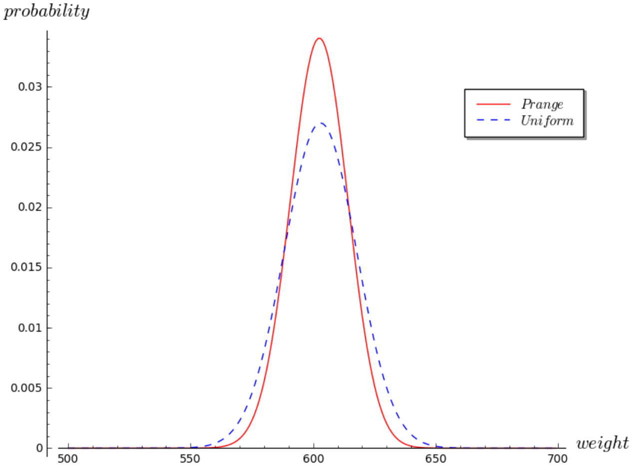
- $\mathbf{e}''_V$ follows a uniform law over $\mathbb{F}_3^{n/2-k_V}$: $\mathbb{E}(|\mathbf{e}''_V|) = \frac{2}{3}(n/2 - k_V)$

- $\mathbf{e}'_V$ can be chosen.

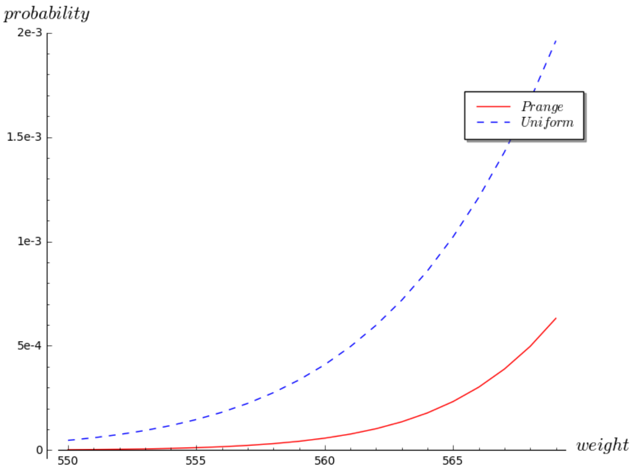
$\longrightarrow$ k_V is fixed as: $\mathbb{E}(|\mathbf{e}'_V|) + \frac{2}{3}(n/2 - k_V) = \mathbb{E}\left(|\mathbf{e}_V^{\text{unif}}|\right)$

REJECTION SAMPLING

Perform rejection sampling!



$$\mathbb{P}(\text{accept}) = \min_j \frac{\mathbb{P}(|\mathbf{e}_V| = j)}{\mathbb{P}(|\mathbf{e}_V^{\text{unif}}| = j)} \ll 1.$$



$$\mathbb{P}(\text{accept}) = \min_j \mathbb{P}(\text{accept}) = \min_j \frac{\mathbb{P}(|\mathbf{e}_V| = j)}{\mathbb{P}(|\mathbf{e}_V^{\text{unif}}| = j)} \ll 1.$$

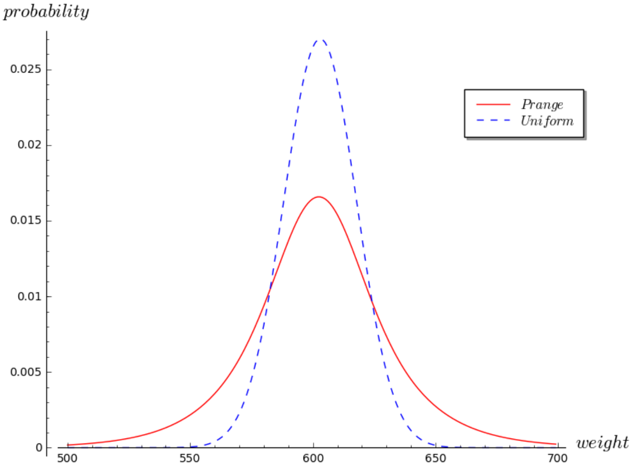
PROBABILISTIC CHOICE OF $\mathbf{e}'_V$

$$\mathbf{e}_V = \underbrace{\boxed{\quad \mathbf{e}'_V \quad}}_{k_V \text{ symbols}}\underbrace{\boxed{\quad \mathbf{e}''_V \quad}}_{n/2 - k_V \text{ symbols}}$$

- $\mathbf{e}''_V$ follows a uniform law: its variance is fixed,

Choose the weight of $\mathbf{e}'_V$ as a random variable!

- $|\mathbf{e}'_V|$ s.t: $\begin{cases} \mathbb{E}(|\mathbf{e}'_V|) + \frac{2}{3}(n/2 - k_V) = \mathbb{E}\left(|\mathbf{e}_V^{\mathrm{unif}}|\right) \\ |\mathbf{e}'_V| \text{ high variance!} \end{cases}$



$$\mathbb{P}(\text{accept}) = \min_j \mathbb{P}(\text{accept}) = \min_j \frac{\mathbb{P}(|\mathbf{e}_V| = j)}{\mathbb{P}(|\mathbf{e}_V^{\text{unif}}| = j)} \approx C^{ste}.$$

REMOVING THE REJECTION SAMPLING

$\longrightarrow$ Distribution $|\mathbf{e}_V|'$ can be **precisely** chosen s.t. $\mathbb{P}(\text{accept}) \approx 1$

Using Renyi divergence argument: **removing rejection sampling**!

Signing algorithm: signatures **don't leak** any information on the secret-key!

REMOVING APPROXIMATION IN PRANGE

To represent $\mathcal{C}$: use a basis/**generator-matrix** $\mathbf{G} \in \mathbb{F}_q^{k \times n}$,

$$\mathcal{C} = \left\{ \mathbf{x}\mathbf{G} \ : \ \mathbf{x} \in \mathbb{F}_q^k \right\} \quad \left(\textit{rows of } \mathbf{G} \textit{ form a basis of } \mathcal{C} \right).$$

Prange algorithm: by linear algebra (Gaussian elimination)

$\mathcal{C}$ has dimension k: $\forall \mathbf{z} \in \mathbb{F}_q^k$, easy to compute $\mathbf{c} \in \mathcal{C}$ such that,

$$\mathbf{c} = \Big[\overbrace{\qquad \mathbf{z} \qquad}^{k \text{ symbols freely chosen}} \Big| \overbrace{\quad /////// \quad}^{n-k,\ \textit{no control}} \Big] \in \mathcal{C}$$

Where the k symbols are picked is not uniform!

1. Pick $\mathcal{I} \subseteq \{1, \cdots, n\}$ such that $\mathbf{G}_{\mathcal{I}}$ has rank k (columns of $\mathbf{G}$ indexed by $\mathcal{I}$),
2. Compute the codeword $\mathbf{x}\mathbf{G}$ where $\mathbf{x} \stackrel{\text{def}}{=} \mathbf{z}\mathbf{G}_{\mathcal{I}}^{-1}$.

NON-UNIFORMITY OF PRANGE

$$\mathbb{P}\left(\mathrm{Prange}(\cdot) = \mathbf{x} \mid |\mathrm{Prange}(\cdot)| = t\right) = \frac{1}{\sharp\{\mathbf{y} : |\mathbf{y}| = t\}} \quad : \text{only } \approx$$

$\longrightarrow$ Only $\approx$ as **we cannot invert** the system for all k coordinates!

$\mathbf{e}_1^{\text{choose}}$	Unif. Distrib.
k columns	$n-k$ symbols

NON-UNIFORMITY OF PRANGE

$$\mathbb{P}\left(\text{Prange}(\cdot) = \mathbf{x} \mid |\text{Prange}(\cdot)| = t\right) = \frac{1}{\sharp\{\mathbf{y} : |\mathbf{y}| = t\}} \quad : \textbf{only} \approx$$

$\longrightarrow$ Only $\approx$ as **we cannot invert** the system for all k coordinates!

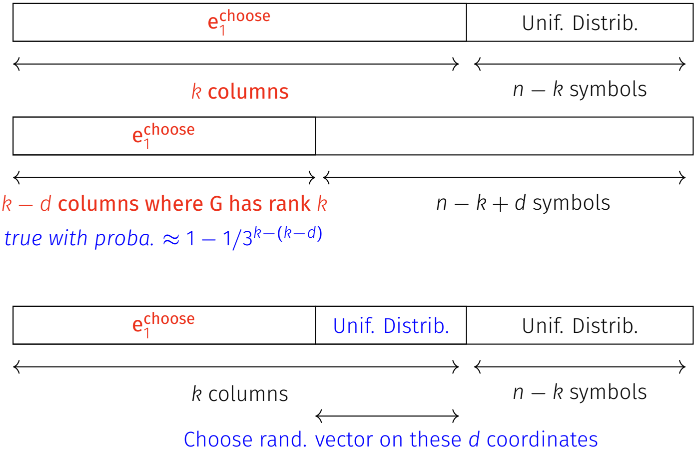
$\mathbf{e}_1^{\text{choose}}$	Unif. Distrib.
$\longleftrightarrow$ k **columns**	$\longleftrightarrow$ $n-k$ symbols

$\mathbf{e}_1^{\text{choose}}$	
$\longleftrightarrow$ $k-d$ **columns where G has rank** k	$\longleftrightarrow$ $n-k+d$ symbols

true with proba. $\approx 1 - 1/3^{k-(k-d)}$

$$\mathbb{P}\left(\text{Prange}(\cdot) = \mathbf{x} \mid |\text{Prange}(\cdot)| = t\right) = \frac{1}{\sharp\{\mathbf{y} : |\mathbf{y}| = t\}} \quad : \text{only } \approx$$

$\longrightarrow$ Only $\approx$ as **we cannot invert** the system for all k coordinates!



Signing algorithm: signatures **don't leak** any information on the secret-key!

$\longrightarrow$ It enables to reduce the security (EUF-CMA in (Q)ROM) to the hardness of:

Security reduction ((Q)ROM):

- Decoding a random linear code at distance $w \approx 0.9n$,
- Distinguishing random and generalized $(U \mid U+V)$-codes.