Variations on the Knapsack Generator

Florette Martinez

Université Picardie - École d'ingénieurs Jules Verne

May 16th, at Rennes

KNAPSACK GENERATOR

KNAPSACK GENERATOR

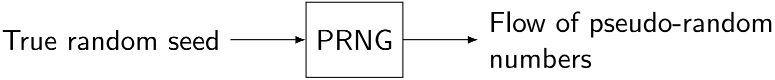
Pseudo Random Number Generator

KNAPSACK GENERATOR

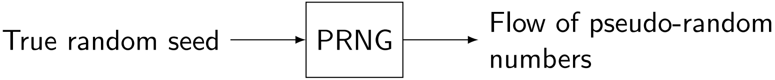
Hard computational problem → KNAPSACK

Pseudo Random Number Generator → GENERATOR

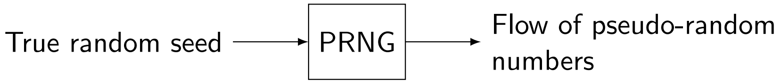
Table of Contents


True random seed
PRNG
Flow of pseudo-random numbers

PRNG



- Security is often based on perfect randomness
- "True" randomness is expensive



- Security is often based on perfect randomness
- "True" randomness is expensive

- Shared randomness is common in cryptography
- Stream cipher
- Reducing communication in MPC protocols.

A PRNG is weak if :

- The flow is not indistinguishable from true randomness

A PRNG is weak if :

- The flow is not indistinguishable from true randomness
- Worse, further outputs are **predictable**

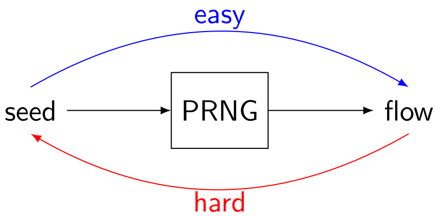
Security of PRNG

A PRNG is weak if :

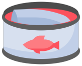
- The flow is not indistinguishable from true randomness
- Worse, further outputs are **predictable**
- Even worse, we can **retrieve the seed** from a reasonable number of outputs.

A PRNG is weak if :

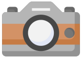
- The flow is not indistinguishable from true randomness
- Worse, further outputs are **predictable**
- Even worse, we can **retrieve the seed** from a reasonable number of outputs.



t kg



ω_1 kg



ω_2 kg

ω_3 kg

ω_4 kg

What is in the knapsack ?

Mathematatic version

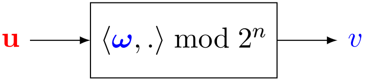
The weight list:	$\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n) \in \{0, N\}^n$
The secret composition:	$\mathbf{u} = (u_1, \ldots, u_n) \in \{0, 1\}^n$
The target weight:	$v = \sum \omega_i u_i = \langle \boldsymbol{\omega}, \mathbf{u} \rangle$

Mathematic version

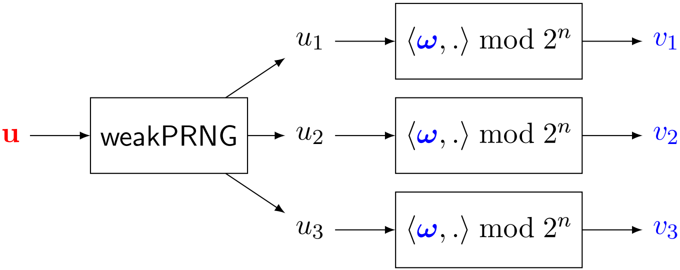
The weight list:	$\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n) \in \{0, N\}^n$
The secret composition:	$\mathbf{u} = (u_1, \ldots, u_n) \in \{0, 1\}^n$
The target weight:	$v = \sum \omega_i u_i = \langle \boldsymbol{\omega}, \mathbf{u} \rangle$

The Subset Sum Problem is NP-hard and remain hard if we replace v by $v \mod N$ as long as $N \simeq 2^n$.

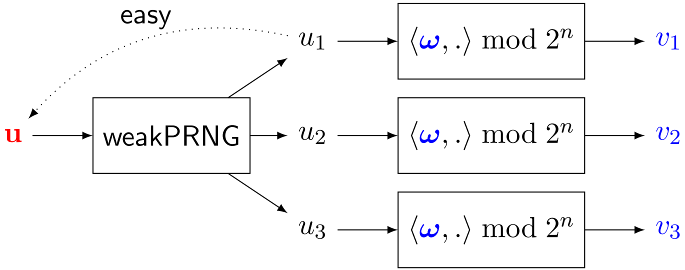
Knapsack Generator by Rueppel and Massey[1]



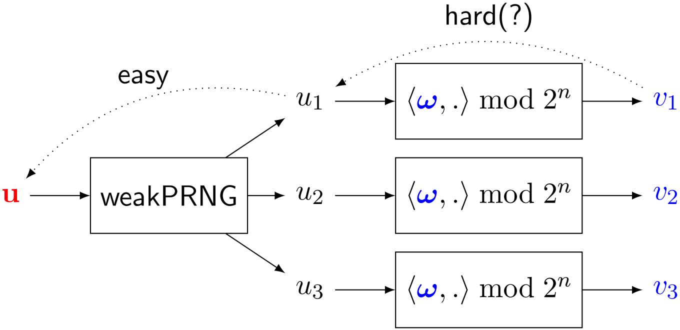
Knapsack Generator by Rueppel and Massey[1]

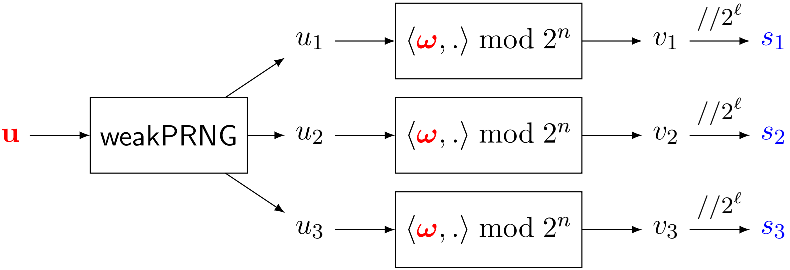


Knapsack Generator by Rueppel and Massey[1]

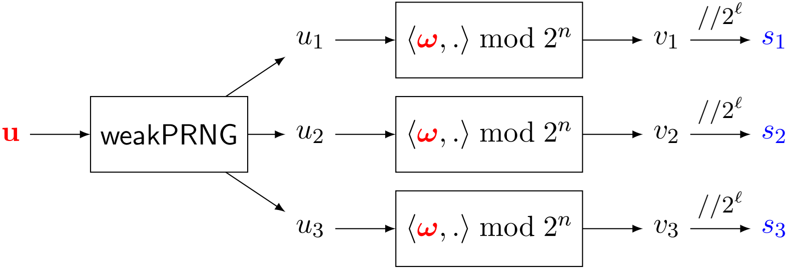


Knapsack Generator by Rueppel and Massey[1]





[1] Rueppel, R.A., Massey, J.L.: Knapsack as a nonlinear function. In: IEEE Intern. Symp. of Inform. Theory, vol. 46 (1985)



We call δ_i the truncated bits : $v_i = 2^\ell s_i + \delta_i$.

[1] Rueppel, R.A., Massey, J.L.: Knapsack as a nonlinear function. In: IEEE Intern. Symp. of Inform. Theory, vol. 46 (1985)

secret : $\mathsf{u} \quad + \quad \omega$

secret :	u	$+$	$\boldsymbol{\omega}$
	n bits		n^2 bits

secret :	u	$+$	$\boldsymbol{\omega}$
	n bits		n^2 bits
	32 bits		1024 bits

secret :	u	$+$	$\boldsymbol{\omega}$
	n bits		n^2 bits
	32 bits		1024 bits

Can we distinguish between the u ?

secret :	u	$+$	$\boldsymbol{\omega}$
	n bits		n^2 bits
	32 bits		1024 bits

Can we distinguish between the u ?
Yes, with OmegaRetriever

We consider m outputs and $\mathbf{s} = (s_1, \ldots, s_m)$.

$$\textsc{OmegaRetriever}: \quad \begin{array}{lll} \mathbf{u}, \mathbf{s} & \to \boldsymbol{\omega}' & \text{close to } \boldsymbol{\omega} \\ \mathbf{u}', \mathbf{s} & \to \boldsymbol{\omega}'' & \text{not close to } \boldsymbol{\omega} \end{array}$$

We consider m outputs and $\mathbf{s} = (s_1, \ldots, s_m)$.

$$\textsc{OmegaRetriever}: \quad \begin{array}{lll} \mathbf{u}, \mathbf{s} & \to \boldsymbol{\omega}' & \text{close to } \boldsymbol{\omega} \\ \mathbf{u}', \mathbf{s} & \to \boldsymbol{\omega}'' & \text{not close to } \boldsymbol{\omega} \end{array}$$

$\textsc{KnapsackGen}(u, \boldsymbol{\omega}')$ will be close to $\textsc{KnapsackGen}(u, \boldsymbol{\omega})$.
$\textsc{KnapsackGen}(u', \boldsymbol{\omega}'')$ will be not.

We consider m outputs and a given $\mathbf{u}$.

- $\mathbf{u} \stackrel{wPRNG}{\longrightarrow} u_1, \ldots, u_m$

We consider m outputs and a given $\mathbf{u}$.

- $\mathbf{u} \stackrel{wPRNG}{\longrightarrow} u_1, \ldots, u_m$
- $U = \begin{pmatrix} u_1 \\ \ldots \\ u_m \end{pmatrix}$

We consider m outputs and a given $\mathbf{u}$.

- $\mathbf{u} \stackrel{wPRNG}{\longrightarrow} u_1, \ldots, u_m$
- $U = \begin{pmatrix} u_1 \\ \ldots \\ u_m \end{pmatrix}$
- $\mathbf{v} = U \times \boldsymbol{\omega} \bmod 2^n$

We consider m outputs and a given $\mathbf{u}$.

- $\mathbf{u} \xrightarrow{wPRNG} u_1, \ldots, u_m$
- $U = \begin{pmatrix} u_1 \\ \ldots \\ u_m \end{pmatrix}$

- $\mathbf{v} = U \times \boldsymbol{\omega} \bmod 2^n$
- $\mathbf{v} = 2^\ell \mathbf{s} + \boldsymbol{\delta}$

We consider m outputs and a given $\mathbf{u}$.

- $\mathbf{u} \overset{wPRNG}{\longrightarrow} u_1, \ldots, u_m$
- $U = \begin{pmatrix} u_1 \\ \ldots \\ u_m \end{pmatrix}$

- $\mathbf{v} = U \times \boldsymbol{\omega} \bmod 2^n$
- $\mathbf{v} = 2^\ell \mathbf{s} + \boldsymbol{\delta}$
- $\boldsymbol{\delta}$ is small $(< 2^\ell)$

We consider m outputs and a given $\mathbf{u}$.

- $\mathbf{u} \overset{wPRNG}{\longrightarrow} u_1, \ldots, u_m$
- $U = \begin{pmatrix} u_1 \\ \ldots \\ u_m \end{pmatrix}$

- $\mathbf{v} = U \times \boldsymbol{\omega} \bmod 2^n$
- $\mathbf{v} = 2^\ell \mathbf{s} + \boldsymbol{\delta}$
- $\boldsymbol{\delta}$ is small ($< 2^\ell$)

$$\boxed{U \times \boldsymbol{\omega} \equiv 2^\ell \mathbf{s} + \boldsymbol{\delta} \bmod 2^n}$$

We consider m outputs and a given $\mathbf{u}$.

- $\mathbf{u} \stackrel{wPRNG}{\longrightarrow} u_1, \ldots, u_m$
- $U = \begin{pmatrix} u_1 \\ \ldots \\ u_m \end{pmatrix}$

- $\mathbf{v} = U \times \boldsymbol{\omega} \bmod 2^n$
- $\mathbf{v} = 2^\ell \mathbf{s} + \boldsymbol{\delta}$
- $\boldsymbol{\delta}$ is small ($< 2^\ell$)

$$\boxed{U \times \boldsymbol{\omega} \equiv 2^\ell \mathbf{s} + \boldsymbol{\delta} \bmod 2^n}$$

We construct T such that :

- $TU = Id \bmod 2^n$ (polynomial)

We consider m outputs and a given $\mathbf{u}$.

- $\mathbf{u} \overset{wPRNG}{\longrightarrow} u_1, \ldots, u_m$
- $U = \begin{pmatrix} u_1 \\ \ldots \\ u_m \end{pmatrix}$

- $\mathbf{v} = U \times \boldsymbol{\omega} \bmod 2^n$
- $\mathbf{v} = 2^\ell \mathbf{s} + \boldsymbol{\delta}$
- $\boldsymbol{\delta}$ is small ($< 2^\ell$)

$$\boxed{U \times \boldsymbol{\omega} \equiv 2^\ell \mathbf{s} + \boldsymbol{\delta} \bmod 2^n}$$

We construct T such that :

- $TU = Id \bmod 2^n$ (polynomial)
- T small (implies solving CVPs)

We consider m outputs and a given $\mathbf{u}$.

- $\mathbf{u} \overset{wPRNG}{\longrightarrow} u_1, \ldots, u_m$
- $U = \begin{pmatrix} u_1 \\ \ldots \\ u_m \end{pmatrix}$

- $\mathbf{v} = U \times \boldsymbol{\omega} \bmod 2^n$
- $\mathbf{v} = 2^\ell \mathbf{s} + \boldsymbol{\delta}$
- $\boldsymbol{\delta}$ is small ($< 2^\ell$)

$$\boxed{U \times \boldsymbol{\omega} \equiv 2^\ell \mathbf{s} + \boldsymbol{\delta} \bmod 2^n}$$

We construct T such that :

- $TU = Id \bmod 2^n$ (polynomial)
- T small (implies solving CVPs)

$$\boxed{\boldsymbol{\omega} = T 2^\ell \mathbf{s} + T\boldsymbol{\delta}}$$

We now have

- $\boldsymbol{\delta}$ small
- T small
- $\boldsymbol{\omega} = T2^{\ell}\mathbf{s} + T\boldsymbol{\delta}$

We now have

- $\boldsymbol{\delta}$ small
- T small
- $\boldsymbol{\omega} = T2^{\ell}\mathbf{s} + T\boldsymbol{\delta}$

$$\boxed{\boldsymbol{\omega}' = T2^{\ell}\mathbf{s}}$$

$$\|\boldsymbol{\omega} - \boldsymbol{\omega}'\| \leq \|T\|\|\boldsymbol{\delta}\|$$

OmegaRetriever from FSE 2011 (part 2)

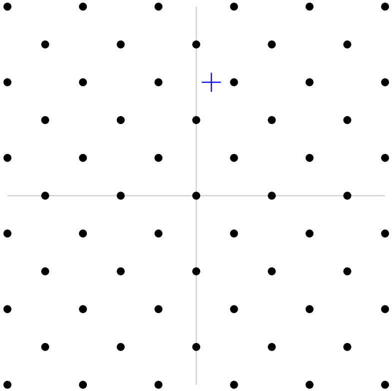
We now have

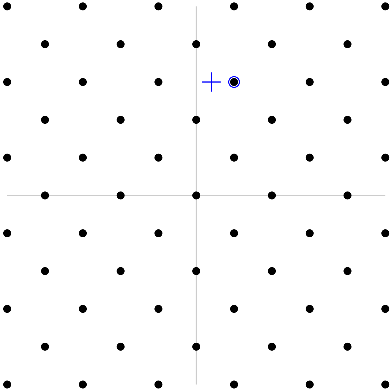
- $\boldsymbol{\delta}$ small
- T small
- $\boldsymbol{\omega} = T2^{\ell}\mathbf{s} + T\boldsymbol{\delta}$

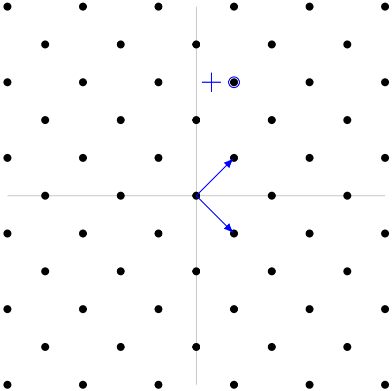
$$\boxed{\boldsymbol{\omega}' = T2^{\ell}\mathbf{s}}$$

$$\|\boldsymbol{\omega} - \boldsymbol{\omega}'\| \leq \|T\|\|\boldsymbol{\delta}\|$$

Experimental results are close to the bound.







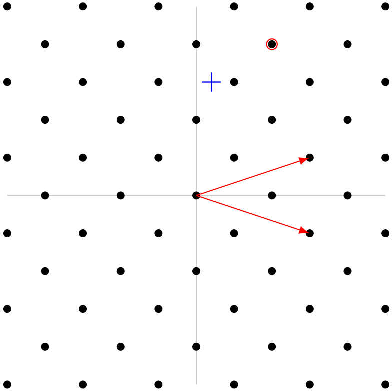


Table of Contents

We consider m outputs and a given $\mathbf{u}$.

1. $\mathbf{v} = U \times \boldsymbol{\omega} \bmod 2^n$
2. $\mathbf{v} = 2^\ell \mathbf{s} + \boldsymbol{\delta}$
3. $\boldsymbol{\delta}$ is small.

We consider m outputs and a given $\mathbf{u}$.

1. $\mathbf{v} = U \times \boldsymbol{\omega} \bmod 2^n$
2. $\mathbf{v} = 2^\ell \mathbf{s} + \boldsymbol{\delta}$
3. $\boldsymbol{\delta}$ is small.

1.	$\longrightarrow$	$\mathbf{v} \in \Lambda$
2. and 3.	$\longrightarrow$	$\mathbf{v}$ is close to $2^\ell \mathbf{s}$

where $\Lambda = \{U \times x \bmod 2^n | x \in \mathbb{Z}^n\}$

We consider m outputs and a given $\mathbf{u}$.

1. $\mathbf{v} = U \times \boldsymbol{\omega} \bmod 2^n$
2. $\mathbf{v} = 2^\ell \mathbf{s} + \boldsymbol{\delta}$
3. $\boldsymbol{\delta}$ is small.

1.	$\longrightarrow$	$\mathbf{v} \in \Lambda$
2. and 3.	$\longrightarrow$	$\mathbf{v}$ is close to $2^\ell \mathbf{s}$

where $\Lambda = \{U \times x \bmod 2^n | x \in \mathbb{Z}^n\}$

$$\mathbf{v}' = \mathsf{CVP}(\Lambda, 2^\ell \mathbf{s})$$

We consider m outputs and a given $\mathbf{u}$.

1. $\mathbf{v} = U \times \boldsymbol{\omega} \bmod 2^n$
2. $\mathbf{v} = 2^\ell \mathbf{s} + \boldsymbol{\delta}$
3. $\boldsymbol{\delta}$ is small.

1.	$\longrightarrow$	$\mathbf{v} \in \Lambda$
2. and 3.	$\longrightarrow$	$\mathbf{v}$ is close to $2^\ell \mathbf{s}$

where $\Lambda = \{U \times x \bmod 2^n | x \in \mathbb{Z}^n\}$

$$\mathbf{v}' = \mathsf{CVP}(\Lambda, 2^\ell \mathbf{s}) \neq \mathbf{v}$$

We consider m outputs and a given $\mathbf{u}$.

1. $\mathbf{v} = U \times \boldsymbol{\omega} \bmod 2^n$
2. $\mathbf{v} = 2^\ell \mathbf{s} + \boldsymbol{\delta}$
3. $\boldsymbol{\delta}$ is small.

1.	$\longrightarrow$	$\mathbf{v} \in \Lambda$
2. and 3.	$\longrightarrow$	$\mathbf{v}$ is close to $2^\ell \mathbf{s}$

where $\Lambda = \{U \times x \bmod 2^n | x \in \mathbb{Z}^n\}$

$$\mathbf{v}' = \mathsf{CVP}(\Lambda, 2^\ell \mathbf{s}) \neq \mathbf{v}$$

But $\boldsymbol{\omega}'$ defined as $U \times \boldsymbol{\omega}' \equiv \mathbf{v}' \bmod 2^n$ is close to $\boldsymbol{\omega}$!

Why does it work ?

- $\mathbf{v} - \mathbf{v}'$ is small and equal to $U \times (\boldsymbol{\omega} - \boldsymbol{\omega}') \bmod 2^n$
- U small because in $\mathcal{M}(\{0,1\})$

- $\mathbf{v} - \mathbf{v}'$ is small and equal to $U \times (\boldsymbol{\omega} - \boldsymbol{\omega}') \bmod 2^n$
- U small because in $\mathcal{M}(\{0, 1\})$
- U small and $\boldsymbol{\omega} - \boldsymbol{\omega}'$ small $\Rightarrow \mathbf{v} - \mathbf{v}'$ small.

- $\mathbf{v} - \mathbf{v}'$ is small and equal to $U \times (\boldsymbol{\omega} - \boldsymbol{\omega}') \bmod 2^n$
- U small because in $\mathcal{M}(\{0,1\})$
- U small and $\boldsymbol{\omega} - \boldsymbol{\omega}'$ small $\Rightarrow \mathbf{v} - \mathbf{v}'$ small.
- U small and $\mathbf{v} - \mathbf{v}'$ small $\not\Rightarrow \boldsymbol{\omega} - \boldsymbol{\omega}'$ small

In the first attack was constructed a small T pseudo inverse of U. Then

In the first attack was constructed a small T pseudo inverse of U. Then

- $\boldsymbol{\omega} - \boldsymbol{\omega}' = T \times (\mathbf{v} - \mathbf{v}') \bmod 2^n$

A First Idea

In the first attack was constructed a small T pseudo inverse of U. Then

- $\boldsymbol{\omega} - \boldsymbol{\omega}' = T \times (\mathbf{v} - \mathbf{v}') \bmod 2^n$
- We can bound T and $(\mathbf{v} - \mathbf{v}')$

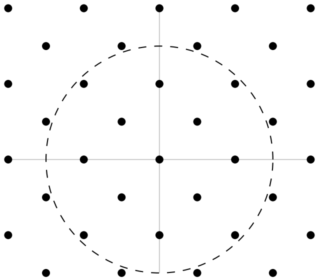
A First Idea

In the first attack was constructed a small T pseudo inverse of U. Then

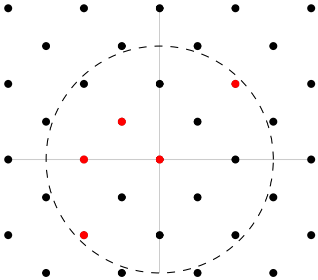
- $\boldsymbol{\omega} - \boldsymbol{\omega}' = T \times (\mathbf{v} - \mathbf{v}') \bmod 2^n$
- We can bound T and $(\mathbf{v} - \mathbf{v}')$
- BUT $\|\boldsymbol{\omega} - \boldsymbol{\omega}'\| \ll \|T\| \times \|(\mathbf{v} - \mathbf{v}')\|$

A Second Idea

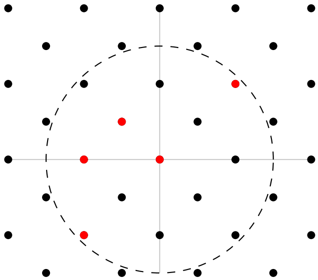
1. We know that $\mathbf{v} - \mathbf{v}'$ is small ($\leq K$) and in Λ.



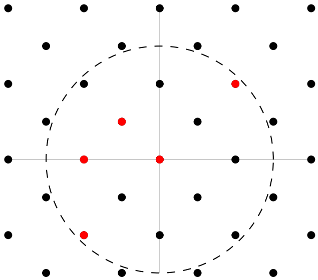
1. We know that $\mathbf{v} - \mathbf{v}'$ is small ($\leq K$) and in Λ.
2. If $\|x\| < K/\|U\|$, then $\|Ux\| < K$.



1. We know that $\mathbf{v} - \mathbf{v}'$ is small ($\leq K$) and in Λ.
2. If $\|x\| < K/\|U\|$, then $\|Ux\| < K$.
3. How do I know that $(\mathbf{v} - \mathbf{v}')$ is a red point ?



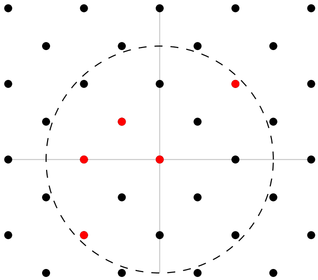
1. We know that $\mathbf{v} - \mathbf{v}'$ is small ($\leq K$) and in Λ.
2. If $\|x\| < K/\|U\|$, then $\|Ux\| < K$.
3. How do I know that $(\mathbf{v} - \mathbf{v}')$ is a red point ?



We denote A_K the set of red points

$$|A_K| = (2 \times \lfloor K/\|U\| \rfloor - 1)^n$$

1. We know that $\mathbf{v} - \mathbf{v}'$ is small ($\leq K$) and in Λ.
2. If $\|x\| < K/\|U\|$, then $\|Ux\| < K$.
3. How do I know that $(\mathbf{v} - \mathbf{v}')$ is a red point ?

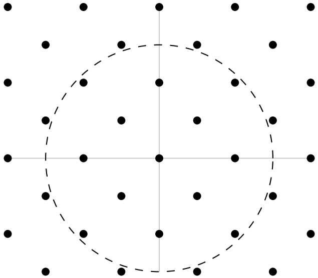


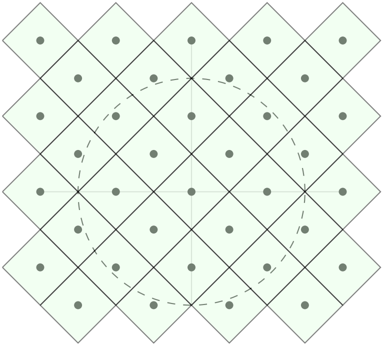
We denote A_K the set of red points

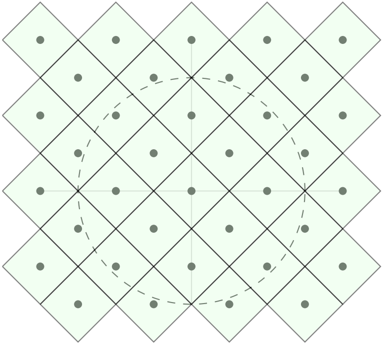
$$|A_K| = (2 \times \lfloor K/\|U\| \rfloor - 1)^n$$

We denote B_K the set of points in the ball

How many point in B_K ?



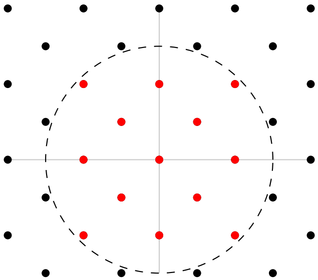




Gaussian Heuristic : $|B_K| \simeq Volume(Ball)/Volume(\Lambda)$

In the case where $n = 32$, $m = 42$ and $\ell \leq 15$,

$$|A_K| \geq |B_K| \text{ with } K = 2^{\ell+1}$$



Thus $\mathbf{v} - \mathbf{v}'$ is a red point and $\|\boldsymbol{\omega} - \boldsymbol{\omega}'\| < K/\|U\|$.

ℓ	5		10		15		20		25	
m	34	40	34	40	34	40	35	40	39	40
✓bits (over 32)	27	28	22	23	5	18	4	13	6	8

Figure: Quality of $\boldsymbol{\omega'}$ for $n = 32$

ℓ	5		10		15		20
m	34	40	35	40	36	40	41
✓bits (over 32)	10	22	10	17	8	12	6

Figure: Quality of $\boldsymbol{\omega'}$ for $n = 32$ for FSE 2011 algorithm

Thank you for your attention,