Ideal Lattices

Vadim Lyubashevsky
Tel-Aviv University

Ideal Lattice FAQs

Q: What are ideal lattices?

A: They are lattices with some additional algebraic structure.

Lattices are groups

Ideal Lattices are ideals

Q: What can we do with ideal lattices?

A: 1. Build efficient cryptographic primitives

2. Build a homomorphic encryption scheme

Cyclic Lattices

A set L in $\mathbf{Z}^n$ is a *cyclic lattice* if:

1.) For all v,w in L, v+w is also in L

-1	2	3	-4	+	-7	-2	3	6	=	-8	0	6	2

2.) For all v in L, -v is also in L

-1	2	3	-4		1	-2	-3	4

3.) For all v in L, a cyclic shift of v is also in L

-1	2	3	-4
-4	-1	2	3
3	-4	-1	2
2	3	-4	-1

Cyclic Lattices = Ideals in $\mathbf{Z}[x]/(x^n-1)$

A set L in $\mathbf{Z}^n$ is a *cyclic lattice* if L is an *ideal* in $\mathbf{Z}[x]/(x^n-1)$

1.) For all v,w in L, v+w is also in L

-1	2	3	-4

+

-7	-2	3	6

=

-8	0	6	2

$(-1+2x+3x^2-4x^3)+(-7-2x+3x^2+6x^3)=(-8+0x+6x^2+2x^3)$

2.) For all v in L, -v is also in L

-1	2	3	-4

1	-2	-3	4

$(-1+2x+3x^2-4x^3)$ $(1-2x-3x^2+4x^3)$

3.) For all v in L, ~~a cyclic shift of v is also in L~~ vx is also in L

-1	2	3	-4	$-1+2x+3x^2-4x^3$
-4	-1	2	3	$(-1+2x+3x^2-4x^3)x=-4-x+2x^2+3x^3$
3	-4	-1	2	$(-1+2x+3x^2-4x^3)x^2 =3-4x-x^2+2x^3$
2	3	-4	-1	$(-1+2x+3x^2-4x^3)x^3 =2+3x-4x^2-x^3$

Why Cyclic Lattices?

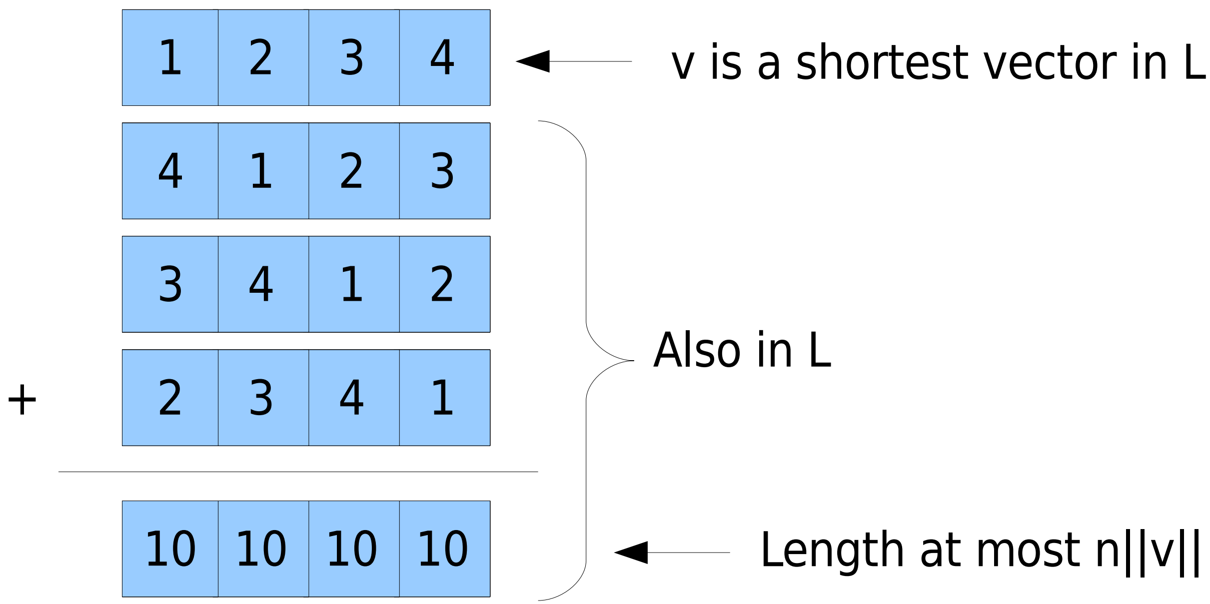
- Succinct representations
  - Can represent an n-dimensional lattice with 1 vector
- Algebraic structure
  - Allows for fast arithmetic (using FFT)
  - Makes proofs possible

- NTRU cryptosystem (fast but no proofs)
- One-way functions based on the worst-case hardness of SVP in cyclic lattices [Mic02]

Is $SVP_{poly(n)}$ Hard for Cyclic Lattices?

Short answer: we don't know but conjecture it is.

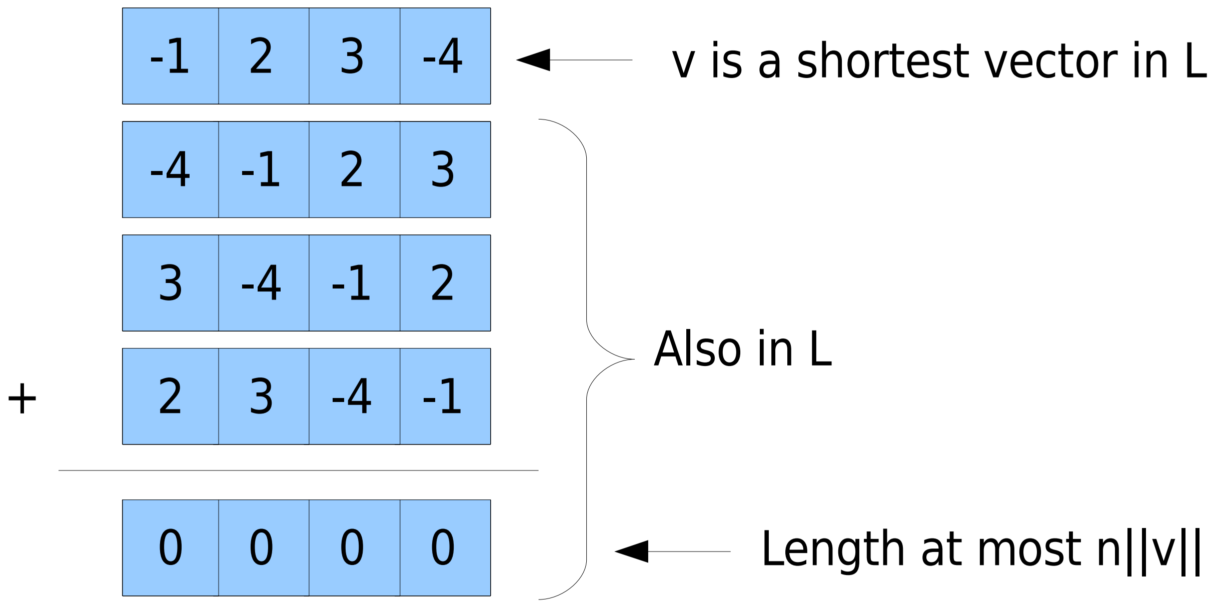
What's wrong with the following argument that SVP_n is easy?

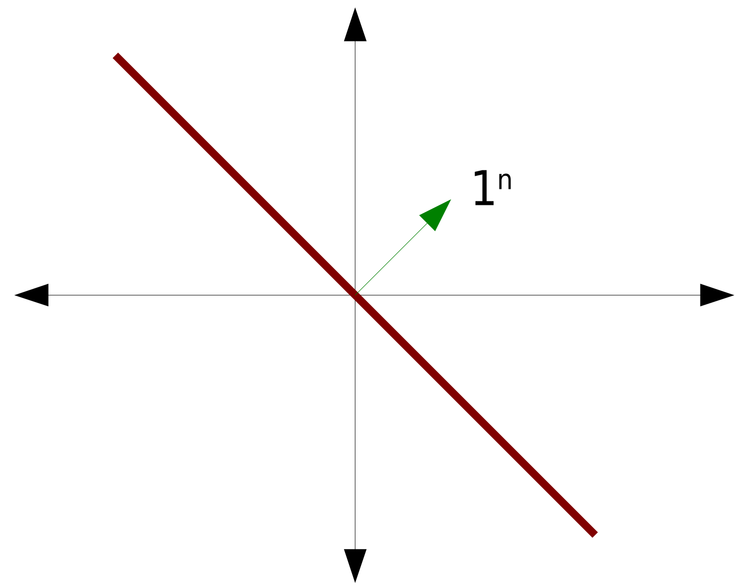


Algorithm for solving $SVP_n(L)$ for a cyclic lattice L:

1. Construct 1-dimensional lattice $L'=L \cap \{1^n\}$
2. Find and output the shortest vector in L'

The Hard Cyclic Lattice Instances





The “hard” instances of cyclic lattices lie on plane P perpendicular to the 1^n vector

In algebra language:

If $R=\mathbf{Z}[x]/(x^n-1)$, then

$$1^n = (x^{n-1}+x^{n-2}+\ldots+1) \approx R/(x-1) \approx \mathbf{Z}[x]/(x-1)$$
$$P = (x-1) \approx R/(x^{n-1}+x^{n-2}+\ldots+1) \approx \mathbf{Z}[x]/(x^{n-1}+x^{n-2}+\ldots+1)$$

f-Ideal Lattices = Ideals in $\mathbf{Z}[x]/(f)$

Want f to have 3 properties:

1) Monic (i.e. coefficient of largest exponent is 1)
2) Irreducible over $\mathbf{Z}$
3) For all polynomials g,h $||gh \bmod f|| < \text{poly}(n)||g||*||h||$

Conjecture: For all f that satisfy the above 3 properties, solving $SVP_{poly(n)}$ for ideals in $\mathbf{Z}[x]/(f)$ takes time $2^{\Omega(n)}$.

Some "good" f to use:

$f=x^{n-1}+x^{n-2}+\ldots+1$ where n is prime

$f=x^n+1$ where n is a power of 2

(x^n+1)-Ideal Lattices = Ideals in $\mathbf{Z}[x]/(x^n+1)$

A set L in $\mathbf{Z}^n$ is an *(x^n+1)-ideal lattice* if L is an *ideal* in $\mathbf{Z}[x]/(x^n+1)$

1.) For all v,w in L, v+w is also in L

1	2	3	4	+	-7	-2	3	6	=	-6	0	6	10

$(1+2x+3x^2+4x^3)+(-7-2x+3x^2+6x^3)=(-6+0x+6x^2+10x^3)$

2.) For all v in L, -v is also in L

1	2	3	4		-1	-2	-3	-4

$(1+2x+3x^2+4x^3)$ $(-1-2x-3x^2-4x^3)$

3.) For all v in L, vx is also in L

1	2	3	4	$1+2x+3x^2+4x^3$
-4	1	2	3	$(1+2x+3x^2+4x^3)x= -4+x+2x^2+3x^3$
-3	-4	1	2	$(1+2x+3x^2+4x^3)x^2 = -3-4x+x^2+2x^3$
-2	-3	-4	1	$(1+2x+3x^2+4x^3)x^3 = -2-3x-4x^2+x^3$

Hardness of Problems for General and (x^n+1)-Ideal Lattices

Exact Versions

	General	(x^n+1)-ideal
SVP	NP-hard	?
SIVP	NP-hard	?
GapSVP	NP-hard	?
uSVP	NP-hard	N/A
BDD	NP-hard	?

Poly(n)-approximate Versions

	General	(x^n+1)-ideal
SVP	?	?
SIVP	?	?
GapSVP	?	Easy
uSVP	?	N/A
BDD	?	?

Legend:

?: *No hardness proofs nor sub-exponential time algorithms are known.*

Colored boxes: *Problems are equivalent*

SVP = SIVP

Lemma: If v is a vector in **Z**[x]/(f) where f is a monic, irreducible polynomial of degree n, then

$$v, vx, vx^2, \ldots vx^{n-1}$$

are linearly independent.

Proof: Suppose not. Let v be in **Z**[x] with deg(v) < n, and $a_0,a_1,a_2,\ldots a_{n-1}$ in **Z** such that

$$a_0v+a_1vx+a_2vx^2+\ldots+a_{n-1}vx^{n-1} \bmod f = 0$$

$$v(a_0+a_1x+a_2x^2+\ldots+a_{n-1}x^{n-1}) \bmod f =0$$

$$vw \bmod f =0$$

f is irreducible (also prime), thus either f|v or f|w.

But deg(v), deg(w) < n, so contradiction.

SVP = SIVP

Lemma: If v is a vector in **Z**[x]/(f) where f is a monic, irreducible polynomial of degree n, then

$$v,\ vx,\ vx^2,\ \ldots\ vx^{n-1}$$

are linearly independent.

1	2	3	4	Shortest vector v
-4	1	2	3	vx
-3	-4	1	2	vx^2
-2	-3	-4	1	vx^3

$$||v|| = ||vx|| = ||vx^2|| = ||vx^3||$$

Corollary: A (x^n+1)-ideal lattice cannot have a unique shortest vector.

$GapSVP_{\sqrt{n}}$ is easy

Fact: For all (x^n+1)-ideal lattices L,

$$\det(L)^{1/n} \leq \lambda_1(L) \leq \sqrt{n}\, \det(L)^{1/n}$$

So $\det(L)^{1/n}$ is a $\sqrt{n}$ – approximation of $\lambda_1(L)$

Proof of fact:

1. $\lambda_1(L) \leq \sqrt{n}\, \det(L)^{1/n}$ is Minkowski's theorem.

2. Let v be the shortest vector of L. Define L'=(v).

(i.e. L' is generated by vectors v, vx, vx^2, … vx^{n-1})

L' is a sublattice of L, so we have

$$\det(L) \leq \det(L') \leq ||v||^n = (\ \lambda_1(L)\ )^n$$

Applications of Ideal Lattices

- One-way functions based on SVP [Mic02]
- Collision-resistant hash functions based on SVP [LM06,PR06,LMPR08,ADLMPR08]
- Tighter worst-case to average-case reductions [PR07]
- One-time signatures based on SVP [LM08]
- Almost practical ID and signature schemes based on SVP [Lyu08]
- Fully homomorphic encryption based on BDD [Gen09]
- Encryption schemes based on quantum hardness of SVP [SSTX09]

Collision-Resistant Hash Function

- Collision-resistant hash function [LM06, PR06, LMPR08]
  - Provable security based on worst-case hardness of approximating $SVP_{\tilde{O}(n)}$
  - Function evaluation in $\tilde{O}(n)$ time vs. $\tilde{O}(n^2)$ for general lattices
  - SWIFFTX hash function entered into SHA-3 competition. Efficient in practice. [ADLMPR08]

The Hash Function Family

Choose p to be a number ≈ $O(n^{1.5})$

Choose elements $a_1,\ldots,a_{3\log(n)}$ randomly in $\mathbf{Z_p}[x]/(x^n+1)$

On an input from $\{0,1\}^{3n\log(n)}$:

$$\overbrace{\underbrace{1\;0\;0\;1}_{y_1}}^{n \text{ bits}}\;\overbrace{\underbrace{0\;1\;0\;1}_{y_2}}^{n \text{ bits}}\;\cdots\;\overbrace{\underbrace{1\;1\;0\;0}_{y_{3\log(n)}}}^{n \text{ bits}} \quad \text{in } \mathbf{Z_p}[x]/(x^n+1)$$

Output: $a_1y_1+a_2y_2+\ldots+a_{3\log(n)}y_{3\log(n)} \bmod p$

Function maps <u>3nlog(n)</u> bits to $\log(p^n)=n\log(p)=$<u>1.5nlog(n)</u> bits

Efficiency of the Hash Function

- The hash function is defined by O(log(n)) elements in $\mathbf{Z}_p[x]/(x^n+1)$
  - Each element requires nlog(p) bits
  - Total space needed $O(n\log^2 n)$ bits
- Computing $a_1y_1+a_2y_2+\ldots+a_{3\log(n)}y_{3\log(n)}$ requires
  - 3log(n) additions: $O(n\log^2 n)$ time
  - 3log(n) multiplications: $O(n\log^3 n)$ time using FFT
- In practice
  - Can exploit parallelism
  - Can do a lot of pre-processing for the FFT

Comparison of Lattice Hash Functions

	General Lattices ([Ajt96, ... ,MR07])	(x^n+1)-ideal lattices ([LM06, PR06, LMPR08])
Storage	$\tilde{O}(n^2)$	$\tilde{O}(n)$
Computing Time	$\tilde{O}(n^2)$	$\tilde{O}(n)$
Hardness Assumption	$\mathrm{SIVP}_{\tilde{O}(n)}$ or $\mathrm{GapSVP}_{\tilde{O}(n)}$	(x^n+1)-ideal $\mathrm{SVP}_{\tilde{O}(n)}$
Best Known Attack Time	$2^{\Omega(n)}$	$2^{\Omega(n)}$

Proof of Security

- Finding collisions in a random hash function instance $a_1,\ldots,a_{3\log(n)}$ (for a_i in $\mathbf{Z}_{\mathbf{p}}[x]/(x^n+1)$) is as hard as solving $SVP_{\tilde{O}(n)}$ in any ideal of $\mathbf{Z}[x]/(x^n+1)$
- Proof similar to the one for general lattices
- Proceed in iterations:

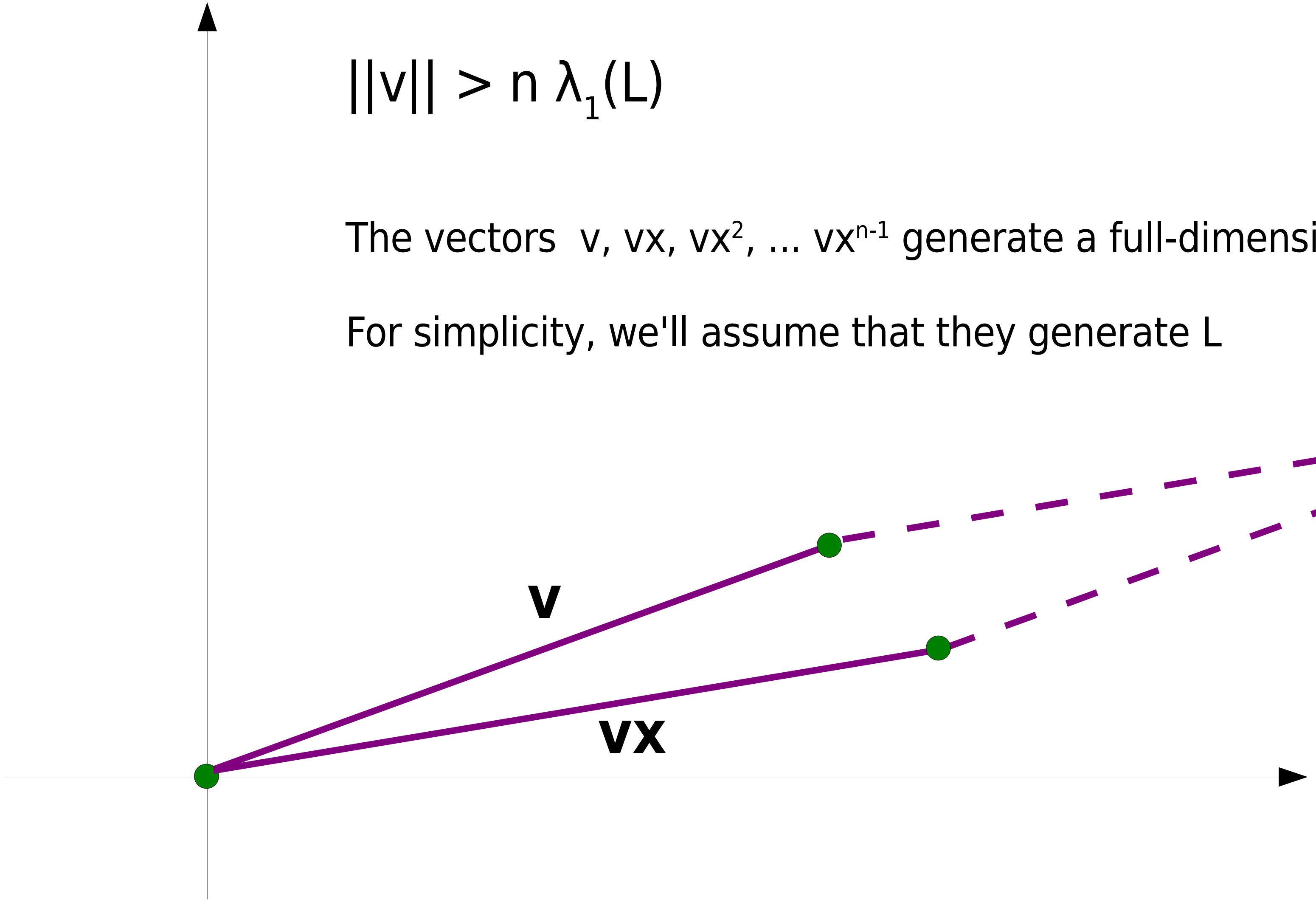
  1) Have some vector in L
  2) Create a random hash function
  3) Finding a collision → finding a shorter vector
  4) Repeat

Security Proof
(From one vector to n vectors)

$||v|| > n\ \lambda_1(L)$

The vectors $v, vx, vx^2, \ldots vx^{n-1}$ generate a full-dimensional sub-lattice of L

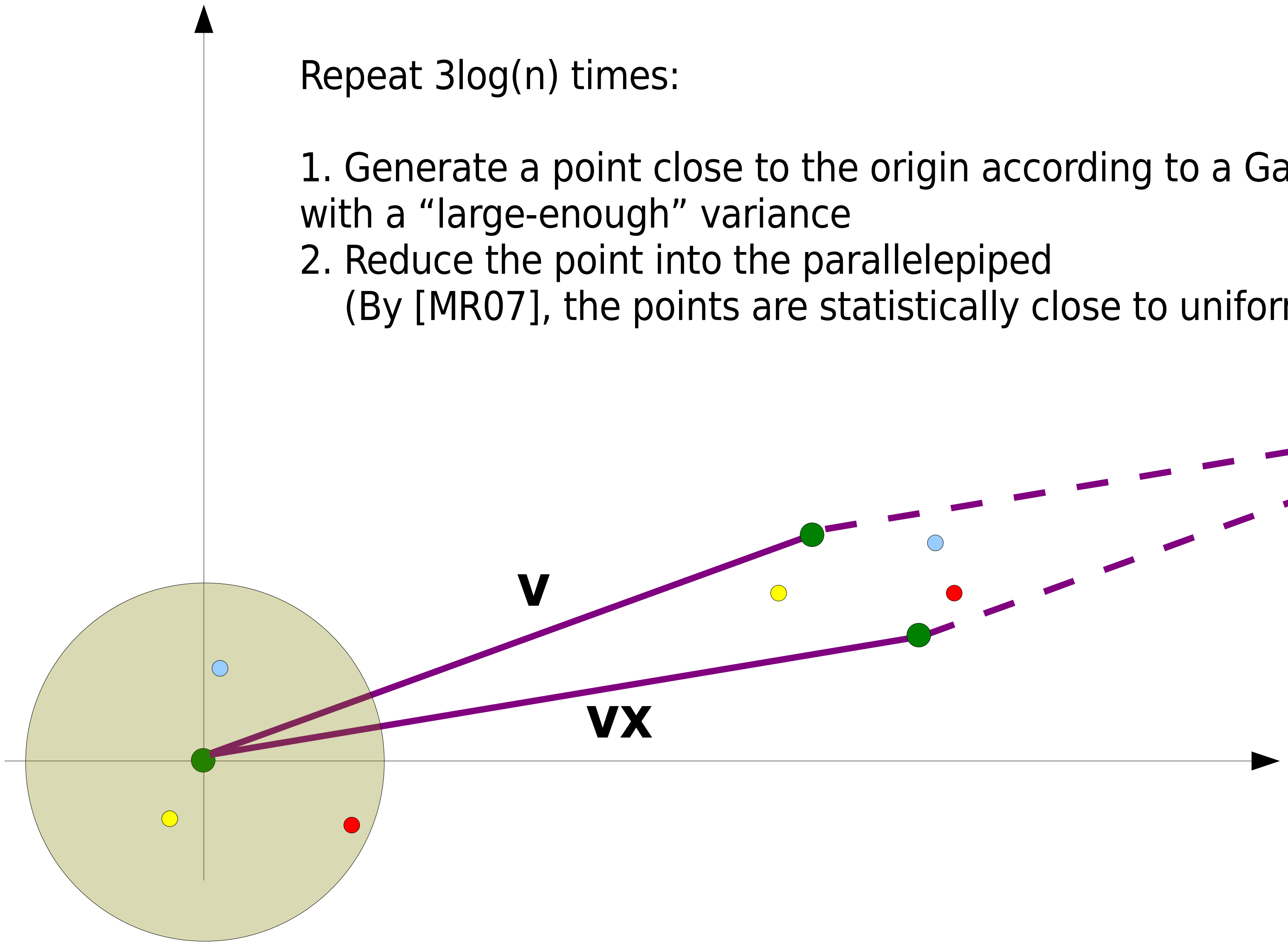
For simplicity, we'll assume that they generate L



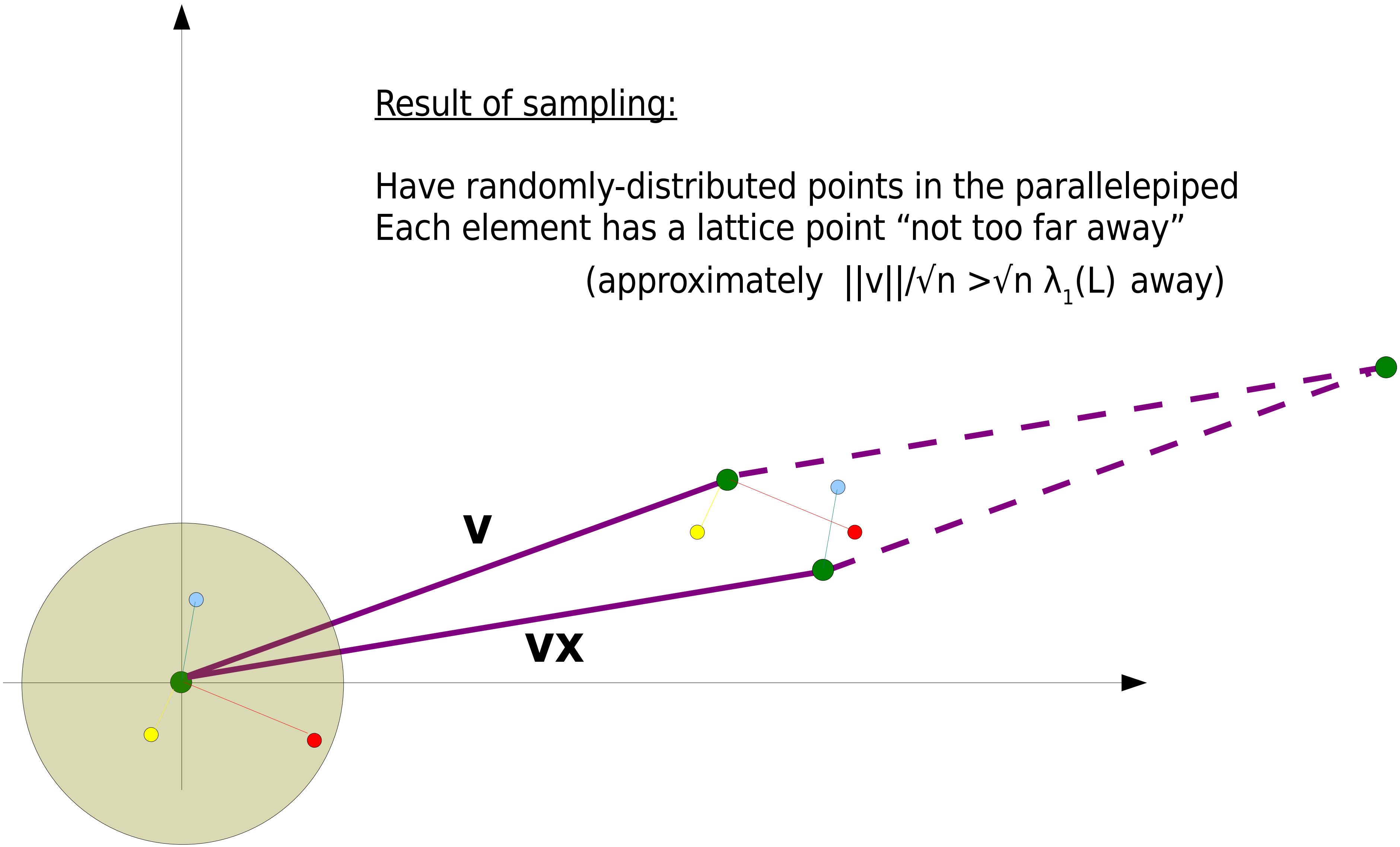
Security Proof
(Getting a random hash function)

Repeat 3log(n) times:

1. Generate a point close to the origin according to a Gaussian distribution with a “large-enough” variance
2. Reduce the point into the parallelepiped
   (By [MR07], the points are statistically close to uniform in $\mathbf{R}^n/L$)



Security Proof
(Getting a random hash function)



Result of sampling:

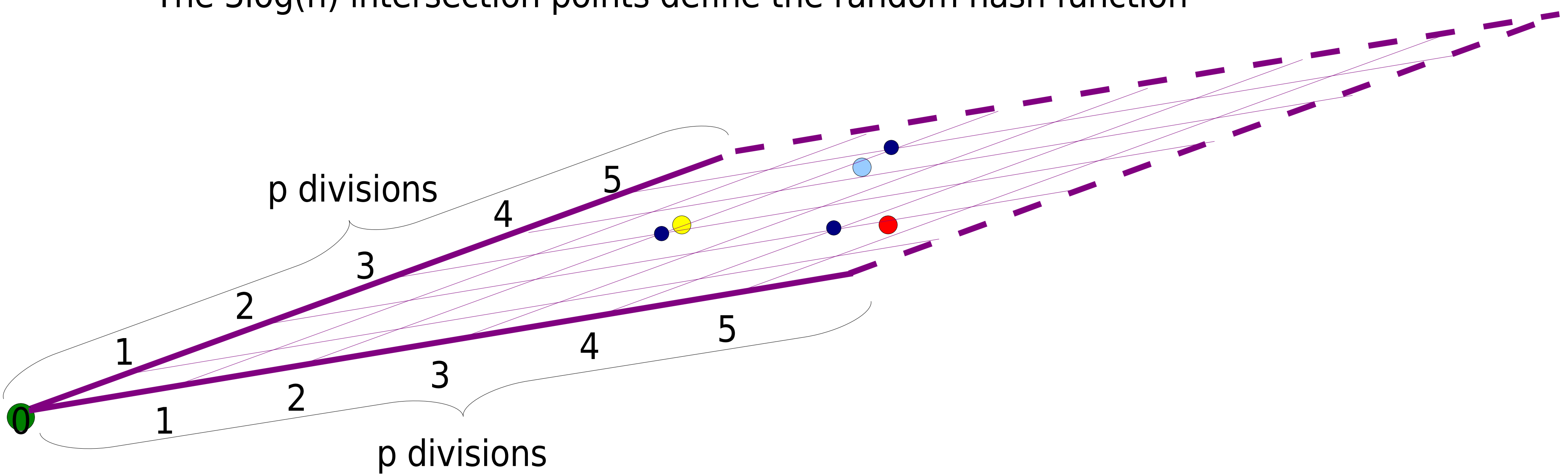
Have randomly-distributed points in the parallelepiped
Each element has a lattice point "not too far away"

(approximately $||v||/\sqrt{n} > \sqrt{n}\,\lambda_1(L)$ away)

Security Proof
(Getting a random hash function)

Subdivide each side of the parallelepiped into p divisions
Each intersection corresponds to an element in $\mathbf{Z}_p[x]/(x^n+1)$

Round each generated point to the nearest intersection
The 3log(n) intersection points define the random hash function



Security Proof

(Finding a collision → finding a shorter vector)

Have a random hash function defined by

$a_1,\ldots,a_{3\log(n)}$

Suppose we find a collision

$$a_1y_1+\ldots+a_{3\log(n)}y_{3\log(n)} = a_1y'_1+\ldots+a_{3\log(n)}y'_{3\log(n)} \bmod p$$

where y_i, y'_i have 0/1 coefficients

Then $a_1(y_1-y'_1)+\ldots+a_{3\log(n)}(y_{3\log(n)} - y'_{3\log(n)}) = 0 \bmod p$

So, $a_1z_1+\ldots+a_{3\log(n)}z_{3\log(n)} = 0 \bmod p$

where z_i have -1/0/1 coefficients

Security Proof
(Finding a collision → finding a shorter vector)

Consider $h = w_1 z_1 + \ldots + w_{3\log(n)} z_{3\log(n)}$ (h is in L because w_i are in L and z_i are in $Z[x]/(x^n+1)$)

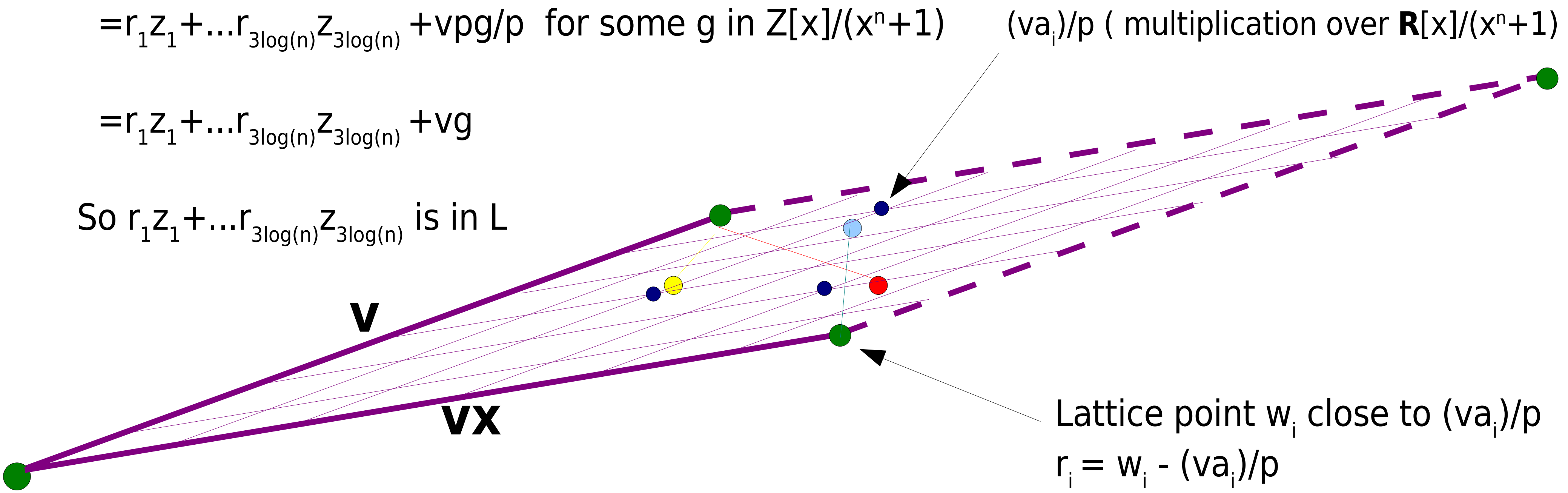
$h = (r_1 + (va_1)/p) z_1 + \ldots + (r_{3\log(n)} + (va_{3\log(n)})/p) z_{3\log(n)}$

$= r_1 z_1 + \ldots r_{3\log(n)} z_{3\log(n)} + v(a_1 z_1 + \ldots + a_{3\log(n)} z_{3\log(n)})/p$

$= r_1 z_1 + \ldots r_{3\log(n)} z_{3\log(n)} + vpg/p$ for some g in $Z[x]/(x^n+1)$

$= r_1 z_1 + \ldots r_{3\log(n)} z_{3\log(n)} + vg$

So $r_1 z_1 + \ldots r_{3\log(n)} z_{3\log(n)}$ is in L



Security Proof

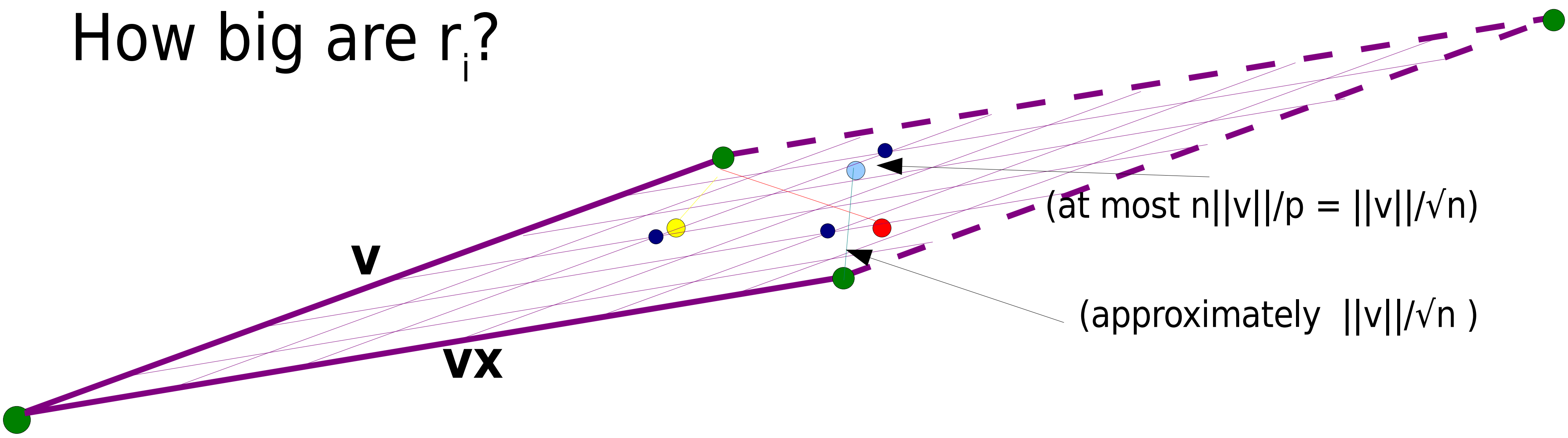
(Finding a collision → finding a shorter vector)

Found a vector $r_1 z_1 + \ldots r_{3\log(n)} z_{3\log(n)}$

How big is it?

z_i have -1/0/1 coefficients (that's small)

How big are r_i?



So $||r_i||$ is on the order of $||v||/\sqrt{n}$

Security Proof

(Finding a collision → finding a shorter vector)

Using the fact that r_i are chosen randomly, and the fact that $||r_i||$ is on the order of $||v||/\sqrt{n}$,

$$||r_1 z_1 + \ldots r_{3\log(n)} z_{3\log(n)}|| = O(||v||)$$

By modifying a few variables by polylog terms, we can make it strictly less than $||v||$

One more thing... need to make sure it's not 0

(Same idea as for general lattices)

One-time Signatures

- Nearly-optimal (asymptotically) 1-time signatures [LM08]
  - Signing and verification takes $\tilde{O}(n)$ time.
  - Breaking signature is conjectured to be $2^{\Omega(n)}$-hard
  - No other such constructions (even ad-hoc) are known
  - A black box conversion from 1-way functions would require $\Omega(n^2)$ time for $2^{\Omega(n)}$-security [BM08]
- Our construction:
  - Based on the hardness of finding collisions in the ideal lattice based hash function
  - Similar in spirit to some number-theoretic constructions

Modules and Hash Functions

Module: Like a *vector space*, but scalars can be in a ring instead of a field

Module $M=(G,R)$

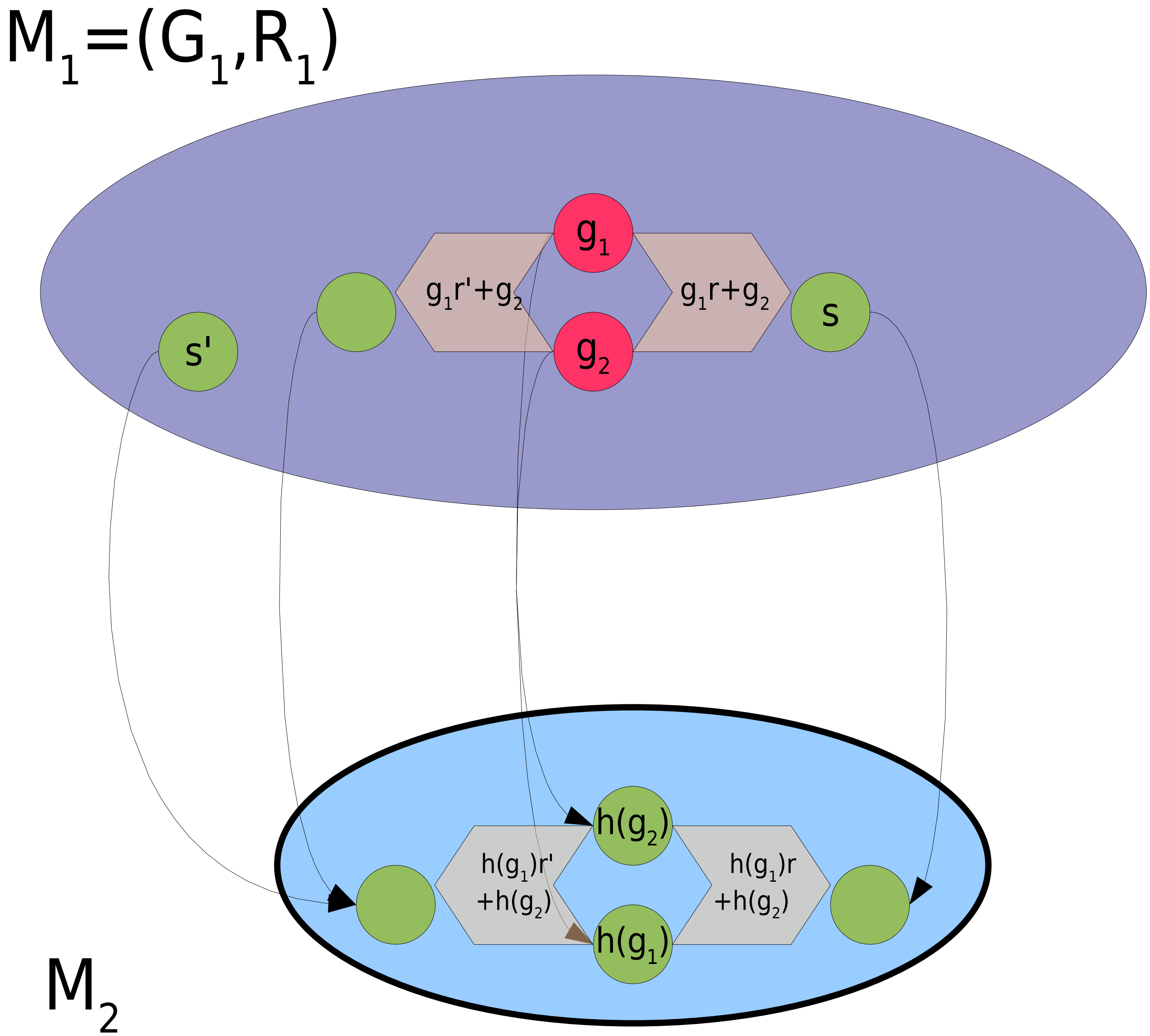
G is an Abelian group. R is a ring.

Module homomorphism $h: M_1 \rightarrow M_2$ satisfies:

- $h(gr)=h(g)r$
- $h(g_1+g_2)=h(g_1)+h(g_2)$

Hardness assumption: hard to find g_1,g_2 such that $h(g_1)=h(g_2)$

One-time Signature Scheme



Generate g_1,g_2 randomly in G_1

Secret Key = (g_1,g_2)

Public Key=$(h(g_1),h(g_2))$

Message r in R_1

Signature of r is $s=g_1r+g_2$

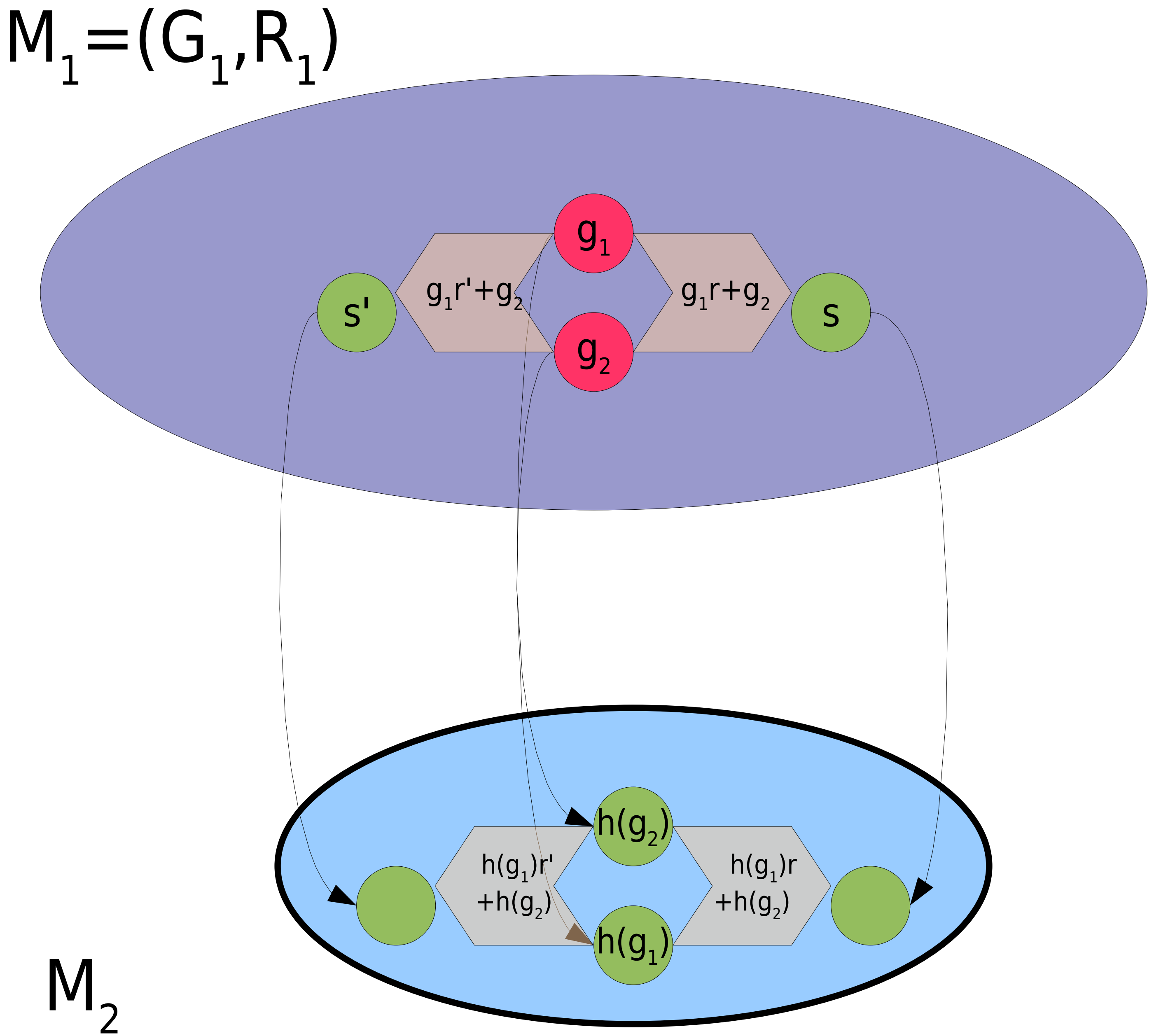
Accept if
$h(s)=h(g_1)r+h(g_2)$

Security proof idea:

Suppose an adversary finds message r' and signature s'

Then we can sign r' and the hash of our signature should equal to h(r')

What if $s'=g_1r'+g_2$?



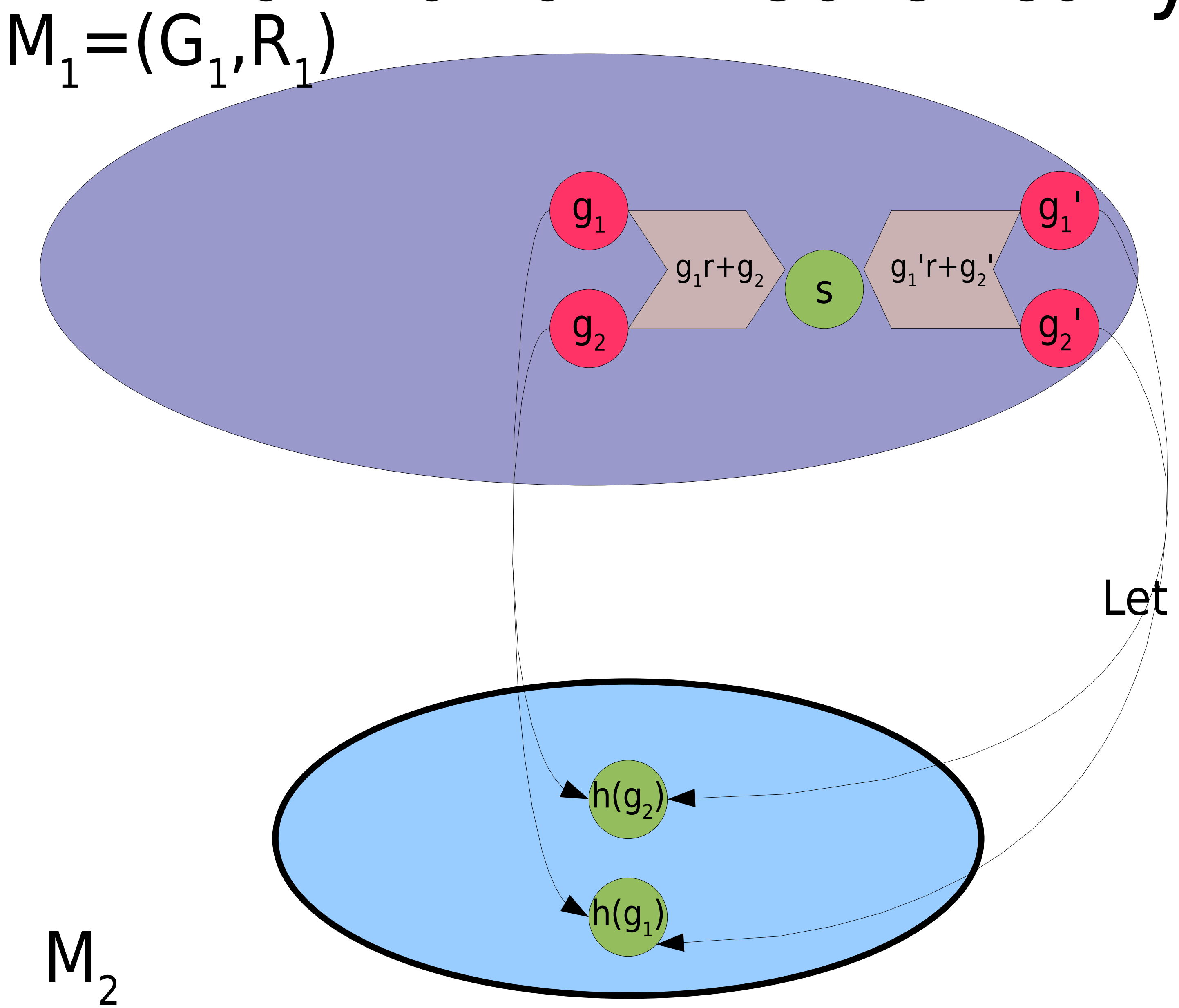
Attacker knows:

$s=g_1r+g_2$
$s'=g_1r'+g_2$

$g_1=(s-s')/(r-r')$
$g_2=s-g_1r$

Not giving us a collision implies knowing g_1 and g_2

g_1 and g_2 are information-theoretically hidden



Attacker knows:

r
$h(g_1)$
$h(g_2)$
$s=g_1r+g_2$

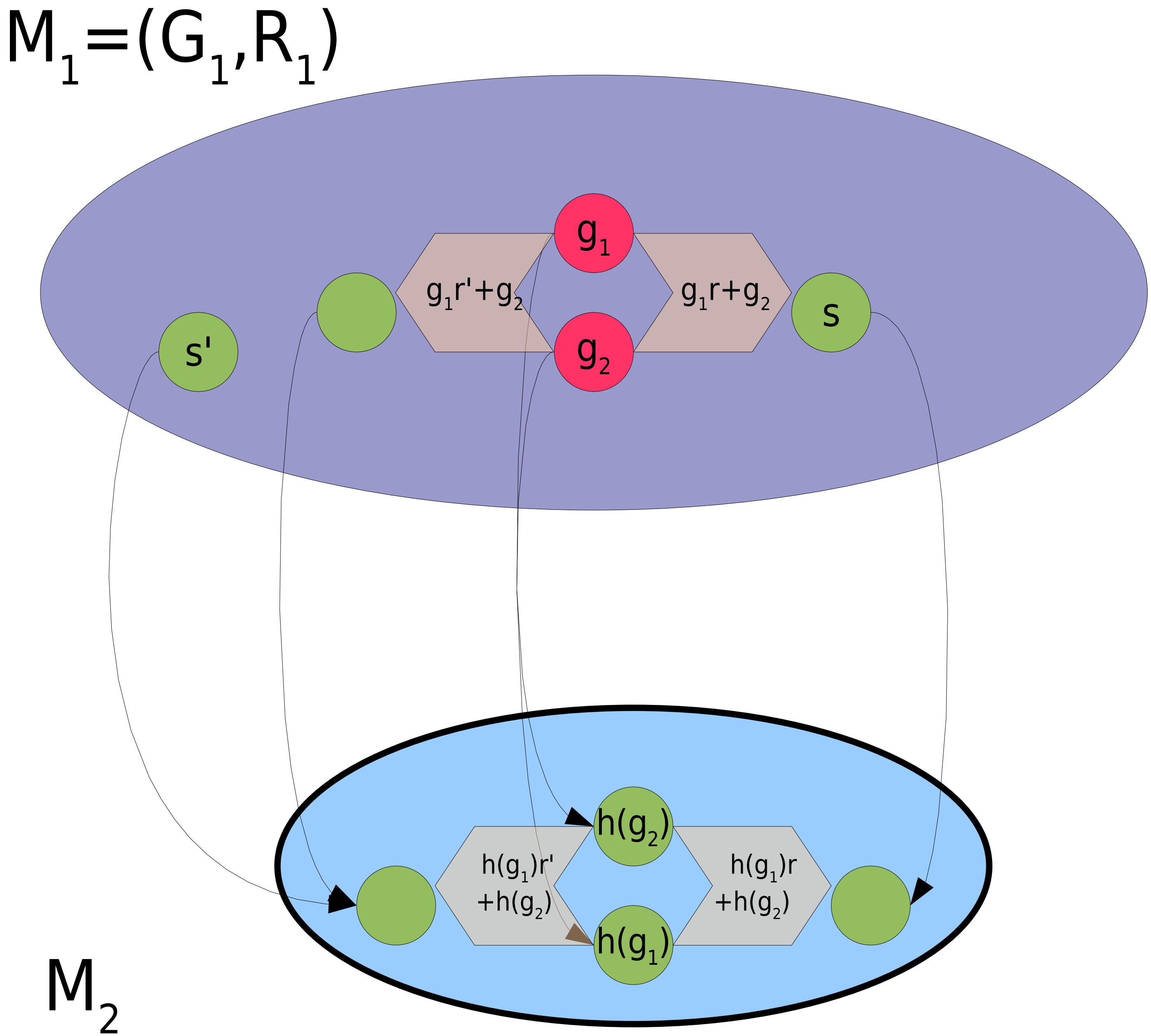
Let z be in the kernel of h
(i.e. $h(z)=0$)

Consider:
$g_1'=g_1-z$
$g_2'=g_2+zr$

Can we do this for ideal lattices?

- $M=(G,R)$
  - $R=\mathbf{Z_p}[x]/(x^n+1)$
  - $G=R^{3\log(n)}$
- $h(y_1,\ldots,y_{3\log(n)})=a_1y_1+\ldots+a_{3\log(n)}y_{3\log(n)} \bmod p$
- Is h collision-resistant?
  - No. It's easy to find $(y_1,\ldots,y_{3\log(n)})$ and $(y'_1,\ldots,y'_{3\log(n)})$ such that $h(y_1,\ldots,y_{3\log(n)})=h(y'_1,\ldots,y'_{3\log(n)})$
  - It's hard to find ***small*** $(y_1,\ldots,y_{3\log(n)})$ and $(y'_1,\ldots,y'_{3\log(n)})$ such that $h(y_1,\ldots,y_{3\log(n)})=h(y'_1,\ldots,y'_{3\log(n)})$

One-time Signature Scheme



Generate *short* g_1,g_2 randomly in G_1

Secret Key = (g_1,g_2)
Public Key=$(h(g_1),h(g_2))$

Message is a *short* r in R_1
Signature of r is $s=g_1r+g_2$

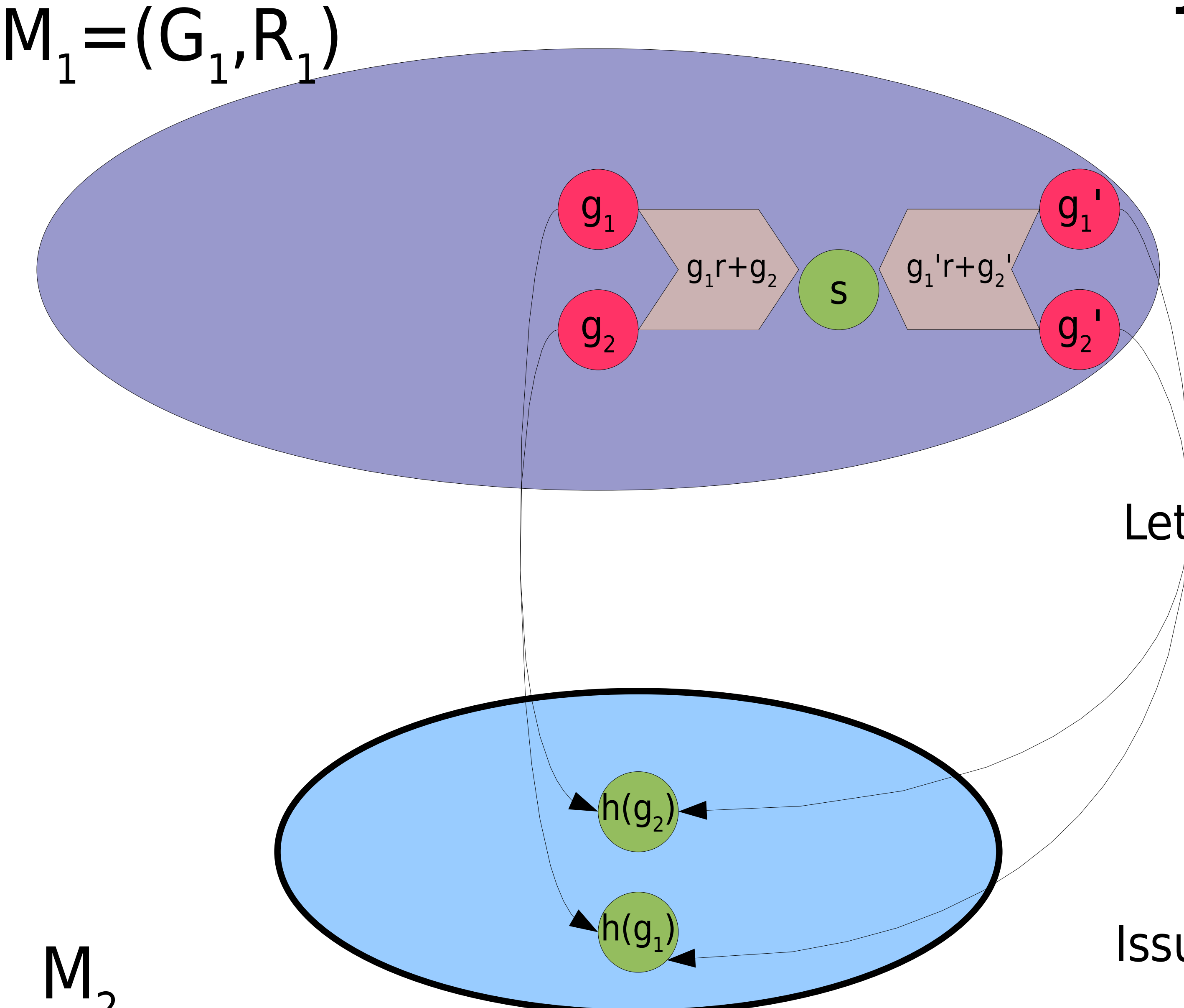
Accept if
$h(s)=h(g_1)r+h(g_2)$ and
s is small

Security proof idea:

Suppose an adversary finds message r' and signature s'

Then we can sign r' and the hash of our signature should equal to h(r')

g_1 and g_2 are information-theoretically hidden



Attacker knows:

r

$h(g_1)$

$h(g_2)$

$s=g_1r+g_2$

Let z be in the kernel of h (i.e. $h(z)=0$)

Consider:

$g_1'=g_1-z$

$g_2'=g_2+zr$

Issue: g_1', g_2' may not be valid secret keys!

Making the lattice scheme work

- Intuitively,
  - Choose secret keys using a distribution such that larger keys are always possible
  - Expected key size is small
  - For any public key and signature, no secret key has too high a prior probability

Some Open Problems

- Design truly practical schemes based on ideal lattices
  - May involve making additional assumptions
- Prove some hardness results for ideal lattice problems
  - If that fails, make up a problem that's hard for ideal lattices
- Prove some non-hardness results for ideal lattice problems
  - e.g. show that SVP_k is not NP hard for $k<\sqrt{n}$
- Show that solving SVP in ideals of $\mathbf{Z}[x]/(f)$ is easy for certain f
  - Might be a good idea to look at f that are "very reducible"
- Does quantum computing help?
  - Ideal lattices have a lot more structure than general lattices
- Design more cryptographic primitives based on ideal lattice problems
  - Almost everything can be done with general lattices. Very few things can be done with ideal lattices