- GGH Public Key Cryptosystem

==== Key Creation ====

1. Choose a good basis $v_1, ..., v_n$.

$\rightarrow$ Choose a set of lineraly independant vectors:

$$v_1, v_2, ..., v_n \in \mathbb{Z}^n$$

- We need them to be reasonably orthogonal. The method I will use is to fix a paramter d and pick random coordinates between $[-d, d]$, we then check how orthogonal they are by computing the Hadamard ratio.

- Let V be the $n-$by$-n$ matrix whose rows are the the vecrots $v_1, ..., v_n$ and let L be the lattice generated by those vectors.

2. Choose an integer matrix U satisfying $det(U) = 1$.

$\rightarrow$ To create U, we take a product of a large number of randomly chosen elementary matrices

3. Compute a bad basis $w_1, ..., w_n$ as the rows of $W = UV$.

4. Publish the public key $w_1, ..., w_n$.

==== Encryption ====

1. Choose small plaintext vector m.

2. Choose random small vector r.

$\rightarrow$ Choose r between $[-\delta, \delta]$, where δ is a fixed public paramter.

3. Use Alice's public key to compute $e = x_1 v_1 + \cdots + x_n v_n + r$.

4. Send the ciphtertext e to Alice.

$$e = mW + r = \sum_{i=1}^{n} m_i w_i + r$$

==== Decryption ====

1. Use Babai's algorithm to compute the vector $v \in L$ closest to e.

2. Compute vW^{-1} to recover m.