Number of solutions: checksum function

MD5 checksums and sizes of the released files:

```
3c63a6d97333f4da35976b6a0755eb67  12732276  Python-3.2.2.tgz
9d763097a13a59ff53428c9e4d098a05  10743647  Python-3.2.2.tar.bz2
3720ce9460597e49264bbb63b48b946d   8923224  Python-3.2.2.tar.xz
f6001a9b2be57ecfbefa865e50698cdf  19519332  python-3.2.2-macosx10.3.dmg
8fe82d14dbb2e96a84fd6fa1985b6f73  16226426  python-3.2.2-macosx10.6.dmg
cccb03e14146f7ef82907cf12bf5883c  18241506  python-3.2.2-pdb.zip
72d11475c986182bcb0e5c91acec45bc  19940424  python-3.2.2.amd64-pdb.zip
ddeb3e3fb93ab5a900adb6f04edab21e  18542592  python-3.2.2.amd64.msi
8afb1b01e8fab738e7b234eb4fe3955c  18034688  python-3.2.2.msi
```

A *checksum function* maps long files to short sequences.

Idea:

- Web page shows the checksum of each file to be downloaded.
- Download the file and run the checksum function on it.
- If result does not match checksum on web page, you know the file has been corrupted.
- If random corruption occurs, how likely are you to detect it?

Impractical but instructive checksum function:

- *input:* an n-vector $\mathbf{x}$ over $GF(2)$
- *output:* $[\mathbf{a}_1 \cdot \mathbf{x}, \mathbf{a}_2 \cdot \mathbf{x}, \ldots, \mathbf{a}_{64} \cdot \mathbf{x}]$

where $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{64}$ are sixty-four n-vectors.

Number of solutions: checksum function

Our checksum function:

- *input:* an n-vector $\mathbf{x}$ over $GF(2)$
- *output:* $[\mathbf{a}_1 \cdot \mathbf{x}, \mathbf{a}_2 \cdot \mathbf{x}, \ \ldots, \ \mathbf{a}_{64} \cdot \mathbf{x}]$

where $\mathbf{a}_1, \mathbf{a}_2, \ \ldots, \ \mathbf{a}_{64}$ are sixty-four n-vectors.

Suppose $\mathbf{p}$ is the original file, and it is randomly corrupted during download.

What is the probability that the corruption is undetected?

The checksum of the original file is $[\beta_1, \ldots, \beta_{64}] = [\mathbf{a}_1 \cdot \mathbf{p}, \ldots, \mathbf{a}_{64} \cdot \mathbf{p}]$.

Suppose corrupted version is $\mathbf{p} + \mathbf{e}$.

Then checksum of corrupted file matches checkum of original if and only if

$$\begin{array}{rcl} \mathbf{a}_1 \cdot (\mathbf{p} + \mathbf{e}) & = & \beta_1 \\ & \vdots & \\ \mathbf{a}_{64} \cdot (\mathbf{p} + \mathbf{e}) & = & \beta_{64} \end{array} \quad \text{iff} \quad \begin{array}{rcl} \mathbf{a}_1 \cdot \mathbf{p} - \mathbf{a}_1 \cdot (\mathbf{p} + \mathbf{e}) & = & 0 \\ & \vdots & \\ \mathbf{a}_{64} \cdot \mathbf{p} - \mathbf{a}_{64} \cdot (\mathbf{p} + \mathbf{e}) & = & 0 \end{array} \quad \text{iff} \quad \begin{array}{rcl} \mathbf{a}_1 \cdot \mathbf{e} & = & 0 \\ & \vdots & \\ \mathbf{a}_{64} \cdot \mathbf{e} & = & 0 \end{array}$$

iff $\mathbf{e}$ is a solution to the homogeneous linear system $\mathbf{a}_1 \cdot \mathbf{x} = 0, \ \ldots \ \mathbf{a}_{64} \cdot \mathbf{x} = 0$.

Number of solutions: checksum function

Suppose corrupted version is $\mathbf{p} + \mathbf{e}$.
Then checksum of corrupted file matches checkum of original if and only if $\mathbf{e}$ is a solution to homogeneous linear system

$$\begin{aligned} \mathbf{a}_1 \cdot \mathbf{x} &= 0 \\ &\vdots \\ \mathbf{a}_{64} \cdot \mathbf{x} &= 0 \end{aligned}$$

If $\mathbf{e}$ is chosen according to the uniform distribution,

$$\begin{aligned} &\text{Probability}\,(\mathbf{p} + \mathbf{e} \text{ has same checksum as } \mathbf{p}) \\ &= \text{Probability}\,(\mathbf{e} \text{ is a solution to homogeneous linear system}) \\ &= \frac{\text{number of solutions to homogeneous linear system}}{\text{number of } n\text{-vectors}} \\ &= \frac{\text{number of solutions to homogeneous linear system}}{2^n} \end{aligned}$$

Question:
How to find out number of solutions to a homogeneous linear system over $GF(2)$?