Vectors over $GF(2)$

Addition of vectors over $GF(2)$:

$$\begin{array}{rccccc} & 1 & 1 & 1 & 1 & 1 \\ + & 1 & 0 & 1 & 0 & 1 \\ \hline & 0 & 1 & 0 & 1 & 0 \end{array}$$

For brevity, in doing $GF(2)$, we often write 1101 instead of [1,1,0,1].

Example: Over $GF(2)$, what is $1101 + 0111$?
Answer: 1010

Vectors over $GF(2)$: Perfect secrecy

Represent encryption of n bits by addition of n-vectors over $GF(2)$.

Example:

Alice and Bob agree on the following 10-vector as a key:

$$\mathbf{k} = [0, 1, 1, 0, 1, 0, 0, 0, 0, 1]$$

Alice wants to send this message to Bob:

$$\mathbf{p} = [0, 0, 0, 1, 1, 1, 0, 1, 0, 1]$$

She encrypts it by adding $\mathbf{p}$ to $\mathbf{k}$:

$$\mathbf{c} = \mathbf{k} + \mathbf{p} = [0, 1, 1, 0, 1, 0, 0, 0, 0, 1] + [0, 0, 0, 1, 1, 1, 0, 1, 0, 1]$$

$$\mathbf{c} = [0, 1, 1, 1, 0, 1, 0, 1, 0, 0]$$

When Bob receives $\mathbf{c}$, he decrypts it by adding $\mathbf{k}$:

$$\mathbf{c} + \mathbf{k} = [0, 1, 1, 1, 0, 1, 0, 1, 0, 0] + [0, 1, 1, 0, 1, 0, 0, 0, 0, 1] = [0, 0, 0, 1, 1, 1, 0, 1, 0, 1]$$

which is the original message.

Vectors over $GF(2)$: Perfect secrecy

If the key is chosen according to the uniform distribution, encryption by addition of vectors over $GF(2)$ achieves *perfect secrecy*.
For each plaintext $\mathbf{p}$, the function that maps the key to the cyphertext

$$\mathbf{k} \mapsto \mathbf{k} + \mathbf{p}$$

is invertible

Since the key $\mathbf{k}$ has the uniform distribution,
the cyphertext $\mathbf{c}$ also has the uniform distribution.

Vectors over $GF(2)$: All-or-nothing secret-sharing using $GF(2)$

- I have a secret: the midterm exam.
- I've represented it as an n-vector $\mathbf{v}$ over $GF(2)$.
- I want to provide it to my TAs Alice and Bob (A and B) so they can administer the midterm while I take vacation.
- One TA might be bribed by a student into giving out the exam ahead of time, so I don't want to simply provide each TA with the exam.
- *Idea:* Provide pieces to the TAs:
  - the two TAs can jointly reconstruct the secret, but
  - neither of the TAs all alone gains any information whatsoever.
- *Here's how:*
  - I choose a random n-vector $\mathbf{v}_A$ over $GF(2)$ randomly according to the uniform distribution.
  - I then compute

$$\mathbf{v}_B := \mathbf{v} - \mathbf{v}_A$$

  - I provide Alice with $\mathbf{v}_A$ and Bob with $\mathbf{v}_B$, and I leave for vacation.

Vectors over $GF(2)$: All-or-nothing secret-sharing using $GF(2)$

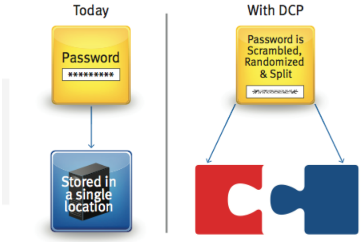
- What can Alice learn without Bob?
- All she receives is a random n-vector.
- What about Bob?
- He receives the output of $f(\mathbf{x}) = \mathbf{v} - \mathbf{x}$ where the input is random and uniform.
- Since $f(\mathbf{x})$ is invertible, the output is also random and uniform.

Vectors over $GF(2)$: All-or-nothing secret-sharing using $GF(2)$

Too simple to be useful, right?
RSA just introduced a product based on this idea:

RSA® DISTRIBUTED CREDENTIAL PROTECTION

Scramble, randomize and split credentials



- Split each password into two parts.
- Store the two parts on two separate servers.

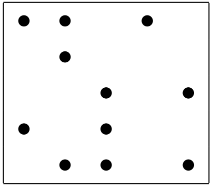
Vectors over $GF(2)$: *Lights Out*

- *input:* Configuration of lights
- *output:* Which buttons to press in order to turn off all lights?

Computational Problem: $\boxed{\text{Solve an instance of } \textit{Lights Out}}$

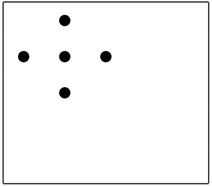
Represent state using `range(5)`$\times$`range(5)`-vector over $GF(2)$.

Example state vector:



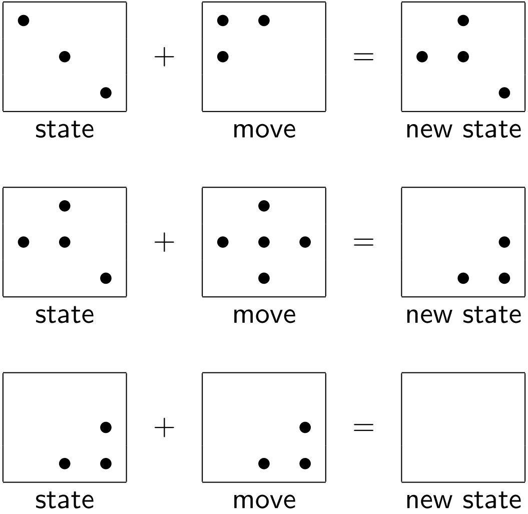
Represent each button as a vector (with ones in positions that the button toggles)

Example button vector:



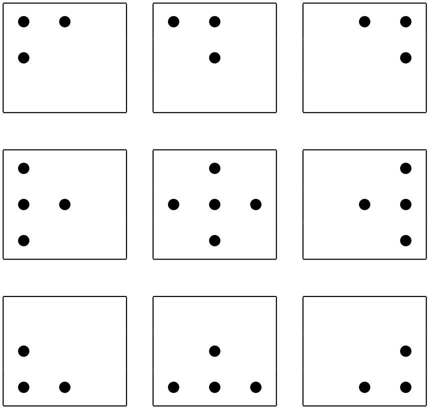
Vectors over $GF(2)$: *Lights Out*

Look at 3×3 case.



Vectors over $GF(2)$: *Lights Out*

Button vectors for 3×3:



Computational Problem: $\boxed{\text{Which sequence of button vectors sum to } \mathbf{s}\text{?}}$

Vectors over $GF(2)$: *Lights Out*

Computational Problem: $\boxed{\text{Which sequence of button vectors sum to } \mathbf{s}\text{?}}$

Observations:

- By commutative property of vector addition, order doesn't matter.
- A button vector occuring twice cancels out.

Replace Computational Problem with: $\boxed{\text{Which } \textit{set} \text{ of button vectors sum to } \mathbf{s}\text{?}}$

Vectors over $GF(2)$: *Lights Out*

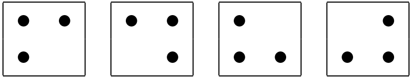
Replace our original Computational Problem with a more general one:

$\boxed{\text{Solve an instance of } \textit{Lights Out}}$ $\Rightarrow$ $\boxed{\text{Which set of button vectors sum to } \mathbf{s}\text{?}}$

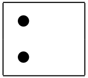
$\Rightarrow$ $\boxed{\text{Find subset of } GF(2) \text{ vectors } \mathbf{v}_1, \ldots, \mathbf{v}_n \text{ whose sum equals } \mathbf{s}}$

Vectors over $GF(2)$: *Lights Out*

Button vectors for 2×2 version:



where the black dots represent ones.

Quiz: Find the subset of the button vectors whose sum is 

Vectors over *GF*(2): *Lights Out*

Button vectors for 2×2 version:

$$\boxed{\begin{array}{cc}\bullet & \bullet\\ \bullet & \end{array}}\quad \boxed{\begin{array}{cc}\bullet & \bullet\\ & \bullet\end{array}}\quad \boxed{\begin{array}{cc}\bullet & \\ \bullet & \bullet\end{array}}\quad \boxed{\begin{array}{cc} & \bullet\\ \bullet & \bullet\end{array}}$$

where the black dots represent ones.

Quiz: Find the subset of the button vectors whose sum is $\boxed{\begin{array}{cc}\bullet & \\ \bullet & \end{array}}$

Answer:

$$\boxed{\begin{array}{cc}\bullet & \\ \bullet & \end{array}} = \boxed{\begin{array}{cc}\bullet & \bullet\\ & \bullet\end{array}} + \boxed{\begin{array}{cc} & \bullet\\ \bullet & \bullet\end{array}}$$