1. Birthday Paradox
2. Relationship to forging a message signature.

If you have 23 or more people in a row, there’s > 50% chance that some pair share a birthday.

\[ n \text{ boxes} \]
\[ \text{Randomly throw } k \text{ balls into the } n \text{ boxes s.t.} \]
\[ \text{each ball has a } \frac{1}{n} \text{ chance of landing in any particular box. What is the probability that at least 1 box has more than 1 ball?} \]

What's probability all different?
\[ \frac{n}{n} \cdot \frac{n-1}{n} \cdot \frac{n-2}{n} \cdot \ldots \cdot \frac{n-k+1}{n} \]

At least 1 box \( \geq 2 \) balls:
\[ 1 - \frac{n!}{n^k \cdot (n-k)!} \]
How to use this idea to get a fake sign.

1. Get a message Person A is willing to sign.

2. Make many different versions of the message by creating words where a synonym could be used, repeating this 16 times so that there are $2^{16}$ roughly equivalent msgs.

3. For each of these call hash func.

<table>
<thead>
<tr>
<th>Choices</th>
<th>HashVal</th>
</tr>
</thead>
<tbody>
<tr>
<td>011010100</td>
<td>1110101.1</td>
</tr>
</tbody>
</table>

$2^{16}$ entries

4. Create your nefarious msg. (One they wouldn't sign but you want to pretend it's from them!)

Create $2^{16}$ versions in the same way.

5. See if any of these versions has a hash value equal to ANYTHING on this constructed table.

6. Get Person A to sign the specific version that matches one of my nefarious messages! But then change the message since hash will be the same!
On average, we need $k = \frac{n}{2}$ for us to expect this technique to work, where $n$ is the output of the hash function. We need $O(2^{\frac{n}{2}})$ memory and $O(2^{\frac{n}{2}})$ time for this technique.