$N = 3$

$2^3 = 8$

$\text{# possible permutations} = 2^3! = 8!$

Key Space = 64-bit, 2^{64} possible keys

Block size = 128 bits, $2^{128}$ possible inputs

$2^{128}!$ possible permutations,

Block cipher defines $2^{64}$ permutation over 128-bit inputs.

Key Space = 8 2-bit long = 00, 01, 10, 11

Block size = 3 bits
BC (01, 001) = 110
\[
\begin{align*}
&\quad \frac{100 \ 011 \ 000}{\text{decrypt w/ key } K'} \\
&\quad \Rightarrow 011 \ 100 \ 111
\end{align*}
\]

attack sees \( K \)

Brute force attack

For \( K = 00, 01, 10, 11 \)

try to decrypt w/ key \( K' \)

Kerchoff Principle: Attacker knows algorithm.

\# Keys = \( 2^{128} \).

DES:

Keys = 56-bits \( \Rightarrow 2^{56} \) keys

Message = 64 bits long \( 2^{64} \) different messages

\# total \# of perms possible \( 2^{64}! \)

DES defines \( 2^{56} \) permutation of 64 bits long
Naive Brute-force attack: try $2^{56}$ keys.
If just 2 rounds long: try $2^{98}$ key to learn one round key.

\[ K \xrightarrow{56} \text{DES} \xrightarrow{64} D \xrightarrow{64} C \]

\[ K_2 \xrightarrow{56} \text{DES}^1 \xrightarrow{64} D \xrightarrow{64} C \]

\[ K_1 \xrightarrow{56} \text{DES} \]

3DES: keysite = 112 bits.
Goal: 112-bits of security.

Want brute force attack to take $2^{112}$ machines, $2^{114}$ trials.

Claim: Break 2DES w/ $\sim 2^{56}$ DES operations.

Input Attack knows $(P_1, C_1), (P_2, C_2)$

Alg:

1. For all keys $K$ in set of 56-bit DES keys
2. Compute $C' = DES(K, P_1)$
3. Store $C'$ in HashTable $Kw/\text{index } C'$

For all keys $K'$ in set of 56-bit DES keys
For all keys $k'$ in set of 56-bit DES keys, compute $c'' = \text{DES}^{-1}(k', c, k)$. Look up $c''$ in Hash Table. If $k' = k_0$, $k_1 = \text{Hash Table}(c'')$.

**Diagram:**

- **Attacker**
  - $M_1$
  - $c_1$
  - $M_2$
  - $c_2$
  - $P_0$, $P_1$
  - $R$

- **Box that encrypts**
  - $c_1 = \text{Enc}(k, M_1)$
  - $c_2 = \text{Enc}(k, M_2)$

- $R = \text{Enc}(k, P_0)$

Outputs $b'$ - its guess for the value $b$. 
Enc scheme secure against this attacker if \( b' = b \) w/ prob close to \( \frac{1}{2} \) (random guessing).

That's why we need random keys.