GMW vs. Yao

- preprocessing?
  * GMW:
    - preprocess all OTs
    - Online phase:
      - does not involve any crypto
      - has low communication
    - round complexity linear in circuit depth
    - overall computation:
      - 1 OT per gate
      - 4-in-1 H-evaluations per gate
        (using load-balancing)
  * Yao:
    - preprocessing
    - preprocess OTs
    - (preprocess garbled-circuit generation)
    - Online phase:
      - garbled-circuit evaluation
      - GC generation
        - high communication
      - round complexity constant
    - overall computation:
      - cost of generating / evaluating garbled gates
      - 3-in-1 AES operations per gate
Semi-honest 2PC implementations:
- Fairplay (2004)
  - Yao
- Tasty (2010)
  - mixed-mode protocols
  - Yao or homomorphic encryption
- Huang et al. (2011)
  - pipelining
  - significantly more efficient / scalable
  - prog. framework
  - Yao
- Choi et al.
  - GMW
- Schneider-Zohner (2013)
  - GMW vs. Yao
TASTY

hom. enc:

given encryption of $a$, $b$
can generate encryption of $a + b$
$[a], [b] \Rightarrow [a+b]$
$[a], r \Rightarrow [r \cdot a]$

Paillier encryption scheme is homomorphic over $\mathbb{Z}_N$

```
S \quad \text{pk} \quad \frac{G}{(pk, sk) \leftarrow \text{Gen}(1^n)}
4 \quad \text{Enc}_{pk}(Y_i) \rightarrow \text{Enc}_{pk}(Y_i)
[a], [b]
```

choose random $r_a, r_b$

$[a + r_a], [b + r_b]$
$[r_b \cdot a], [r_a \cdot b]$

\[ ([a + r_a] \cdot [b + r_b]) \cdot [r_a \cdot b] \cdot [r_b \cdot a] \cdot [-r_a, r_b] \] = $\boxed{[a \cdot b]}$

\[ \frac{[2]}{[z + r_z]} \rightarrow \frac{[z]}{z + r_z} \]
\[ \Gamma \]

\[ \hat{X} : \text{garbled output value} \]

\[ \hat{X} = [(k_1, x_1), \ldots, (k_e, x_e)] \]

\[ \text{Enc}_{\hat{X}}(x_i): \]

For all \( i \) \{ 
\[ \begin{align*} 
& \text{if } \Pi_i = 0 \\
& \quad \text{randomize } C_i \\
& \text{if } \Pi_i = 1 \\
& \quad C_i = C_i^{-1} \\
& \end{align*} \]
\[ \hat{x} = \prod_{i=1}^{e} C_i^{x_i} \]

* conversion between modes is expensive