**Cut & Choose, continued**

**Cut & Choose idea:** [LP07]
- Alice garbles $f$, independently $S$ times
- Bob picks random subset to be "checked" ("check circuits")
- Bob evaluates remaining garbled circuits, outputs majority output value

**Selective failure issue:**
- Alice can make Bob’s probability of abort depend on his input!

**Mitigation from LP07:**

To securely evaluate $f$:
- Securely evaluate $\hat{f}$ instead
- For each input $x_i$ for Bob, pick random $x_{i,1}, \ldots, x_{i,s}$ such that XOR is $x_i$

**Claim:**
$$\left| \Pr[\text{Bob aborts w/ input } x_B] - \Pr[\text{Bob aborts w/ input } \hat{x}_B] \right| \leq \frac{n}{2^{S-1}}$$
where $n$ is # of input bits for Bob
Pf: focus on Bob's first input bit to $f$

Alice provides wire labels for bit $= 0 / 1$

(1) she gives invalid inputs - for both 0/1
   $\Rightarrow$ Bob picks up invalid wire labels regardless of his input

(2) otherwise, for each bit, at most 1 wire label invalid
   Bob's input bit gets expanded into 5 bits in $\hat{f}$

(2a) Alice sends invalid wire label in $< 5$ of these

Bob's distribution over these subset of bits (hence, abort probability)
trap: safe is indep of his "true" input to $f$

(2b) Alice sends invalid wire labels in all 5

Ex: sends invalid wire label for "1" for all of the bits
   (so if Bob has "1" on any of the expanded inputs, he aborts)

   $\Rightarrow$ Bob has input 1 to $f$ $\Rightarrow$ must have some 1
   among expanded inputs
   $\Rightarrow$ aborts w/ prob $= 1$

   $\Rightarrow$ Bob has input 0 to $f$ $\Rightarrow$ aborts w/ prob $= 1 - \frac{1}{2^5}$

Attack: Bob must evaluate many circuits
Alice can send inconsistent inputs: different Alice-input for each garbled circuit
Example: \( f(x, y) = \sum x_i y_i \mod 2 \)

Alice gets Bob to evaluate

majority \{ f(x_1, y), f(x_2, y), \ldots, f(x_k, y) \}

where
\[
\begin{align*}
  x_1 &= 100_0 \ldots \\
  x_2 &= 0100 \ldots \\
  x_3 &= 001 \ldots 
\end{align*}
\]

\[
\begin{align*}
  y_1 &\quad y_2 &\quad \cdots &\quad y_k 
\end{align*}
\]

\[\Rightarrow \text{Bob outputs majority value of } \{ y_1, \ldots, y_k \} \]

**Input Consistency Problem** Alice uses different choices of inputs for different evaluation circuits

**Building Block:** Commitment scheme (locked box)

**Protocol:** Sender has value \( x \)

choose random \( s \leftarrow \mathbb{G}_1 \)

compute \( h := H(s, x) \) \( H \) is hash func, SHA-3

send \( h \)

\[\ldots \text{ later} \ldots\]

send \( s, x \)

receiver can check \( h \stackrel{?}{=} H(s, x) \)

**Security:**
- hard to find \( (s, x) \) \( (s', x') \) s.t. \( H(s, x) = H(s', x') \)
  \( \Rightarrow \) sender is "bound" to single value \( x \)
- for random \( s \), \( H(s, x) \) gives no info about \( x \)
  \( \Rightarrow \) commitment hides \( x \) until opened