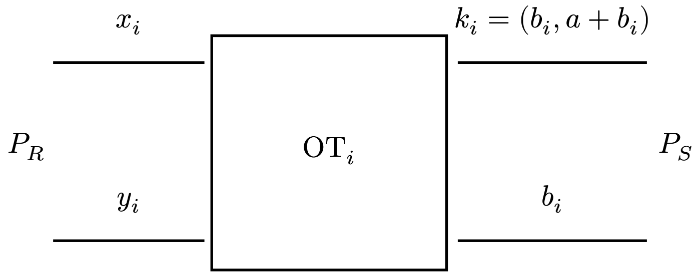
M2A

OT sender and receiver want to get an **additive** sharing $\left(y, \overline{b}\right)$ from a **multiplicative** sharing (x, a). So the sender starts with x and ends up with y and the receiver starts with a and ends up with $\overline{b}$.

They compute $y = ax + b \Leftrightarrow y - b = ax \Leftrightarrow y + \overline{b} = ax$ in n OTs, where n is given by the bitsize of the field elements y, x, a, b. The receiver's inputs for every i-th OT are x_i and his output is y_i. The sender's inputs are a linear combination of a and b_i and he outputs b_i.

Compute $y = ax + b$



The OT sender P_S:

1. Sample n random field elements $b_i \leftarrow \$$ so that $b = \sum_{i=0}^{n} 2^i b_i$
2. In each i-th OT: Send $k_i = (b_i, a + b_i)$ to P_R
3. Compute and output $\overline{b} = -b = -\sum_{i=0}^{n} 2^i b_i$

The OT receiver P_R:

1. Bit-decomposes $x = \sum_{i=0}^{n} 2^i x_i$
2. In each i-th OT: Depending on the bit of x_i he receives $y_i = k_i^{x_i}$, which is
   - $k_i^0 = b_i$ if $x_i = 0$
   - $k_i^1 = a + b_i$ if $x_i = 1$
3. Compute and output $y = \sum_{i=0}^{n} 2^i k_i = ax + b$

Correctness Check

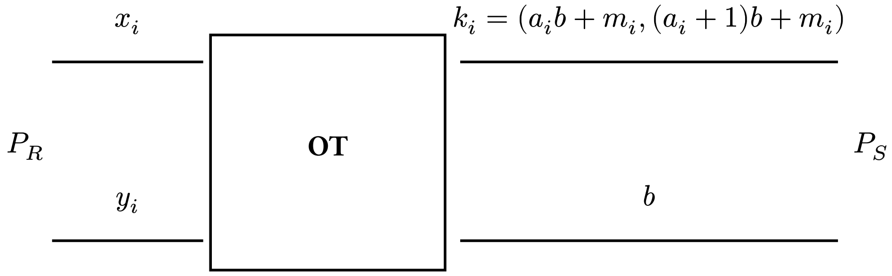
1. Repeat the whole M2A protocol for the same x but with random a_2, b_2. We now label $a_1 := a$ and $b_1 := b$, which are the original values from the previously executed M2A protocol
2. P_R sends 2 random field elements χ_1, χ_2 to P_S
3. P_S computes $a^* = \chi_1 a_1 + \chi_2 a_2$ and $b^* = \chi_1 b_1 + \chi_2 b_2$ and sends them to P_R.
4. P_R checks that $\chi_1 y_1 + \chi_2 y_2 = a^* x + b^*$

A2M

OT sender and receiver want to get a **multiplicative** sharing $\left(y, \overline{b}\right)$ from an **additive** sharing (x, a). So the sender starts with x and ends up with y and the receiver starts with a and ends up with $\overline{b}$.

They compute $y = (a + x)b \Leftrightarrow yb^{-1} = a + x \Leftrightarrow y\overline{b} = a + x$ in n OTs, where n is given by the bitsize of the field elements y, x, a, b. The receiver's inputs for every i-th OT are x_i and his output is y_i. The sender's inputs are a linear combination of a_i and b including a mask m_i and he outputs b.

Compute $y = (a + x)b$



The OT sender P_S:

1. Sample a random field element $b \leftarrow \$$
2. Sample n random field elements $m_i \leftarrow \$$, with $\sum_{i=0}^{n} 2^i m_i = 0$
3. Bit-decomposes $a = \sum_{i=0}^{n} 2^i a_i$
4. In each i-th OT: Send $k_i = (a_i b + m_i, (a_i + 1)b + m_i)$ to P_R
5. Compute and output $\bar{b} = b^{-1}$

The OT receiver P_R:

1. Bit-decomposes $x = \sum_{i=0}^{n} 2^i x_i$
2. In each i-th OT: Depending on the bit of x_i he receives $y_i = k_i^{x_i}$, which is
   - $k_i^0 = a_i b + m_i$ if $x_i = 0$
   - $k_i^1 = (a_i + 1)b + m_i$ if $x_i = 1$
3. Compute and output $y = \sum_{i=0}^{n} 2^i k_i = (a + x)b$

Correctness Check

1. Repeat the whole A2M protocol for the same x but with random a_2, b_2. We now label $a_1 := a$ and $b_1 := b$, which are the original values from the previously executed A2M protocol
2. P_R sends 2 random field elements χ_1, χ_2 to P_S
3. P_S computes $z^* = \chi_1 a_1 b_1 + \chi_2 a_2 b_2$ and $b^* = \chi_1 b_1 + \chi_2 b_2$ and sends them to P_R.
4. P_R checks that $\chi_1 y_1 + \chi_2 y_2 = b^* x + z^*$