$$v \in \mathbb{F}_p,\ w \in \text{String},\ \iota \in \text{Clients} \subset \mathbb{N}$$

$$\begin{array}{lcll}
\varepsilon & ::= & \mathtt{r}[w] \mid \mathtt{s}[w] \mid \mathtt{m}[w] \mid \mathtt{p}[w] \mid & \textit{expressions} \\
 & & v \mid \varepsilon - \varepsilon \mid \varepsilon + \varepsilon \mid \varepsilon * \varepsilon & \\
x & ::= & \mathtt{r}[w]@\iota \mid \mathtt{s}[w]@\iota \mid \mathtt{m}[w]@\iota \mid \mathtt{p}[w] \mid \mathtt{out}@\iota & \textit{variables} \\
\pi & ::= & \mathtt{m}[w]@\iota := \varepsilon@\iota \mid \mathtt{p}[w] := e@\iota \mid \mathtt{out}@\iota := \varepsilon@\iota \mid \pi;\pi & \textit{protocols}
\end{array}$$

$$\begin{array}{rcl}
[\![\sigma, v]\!]_\iota & = & v \\
[\![\sigma, \varepsilon_1 + \varepsilon_2]\!]_\iota & = & [\![[\![\sigma, \varepsilon_1]\!]_\iota + [\![\sigma, \varepsilon_2]\!]_\iota]\!] \\
[\![\sigma, \varepsilon_1 - \varepsilon_2]\!]_\iota & = & [\![[\![\sigma, \varepsilon_1]\!]_\iota - [\![\sigma, \varepsilon_2]\!]_\iota]\!] \\
[\![\sigma, \varepsilon_1 * \varepsilon_2]\!]_\iota & = & [\![[\![\sigma, \varepsilon_1]\!]_\iota * [\![\sigma, \varepsilon_2]\!]_\iota]\!] \\
[\![\sigma, \mathtt{r}[w]]\!]_\iota & = & \sigma(\mathtt{r}[w]@\iota) \\
[\![\sigma, \mathtt{s}[w]]\!]_\iota & = & \sigma(\mathtt{s}[w]@\iota) \\
[\![\sigma, \mathtt{m}[w]]\!]_\iota & = & \sigma(\mathtt{m}[w]@\iota) \\
[\![\sigma, \mathtt{p}[w]]\!]_\iota & = & \sigma(\mathtt{p}[w])
\end{array}$$

$$(\sigma, x := \varepsilon@\iota) \Rightarrow \sigma\{x \mapsto [\![\sigma, \varepsilon]\!]_\iota\} \qquad \frac{(\sigma_1, \varepsilon_1) \Rightarrow \sigma_2 \qquad (\sigma_2, \varepsilon_2) \Rightarrow \sigma_3}{(\sigma_1, \varepsilon_1; \varepsilon_2) \Rightarrow \sigma_3}$$

$$\begin{array}{lcll}
(\sigma, x := \varepsilon@\iota) & \Rightarrow_{\mathcal{A}} & \sigma\{x \mapsto [\![\sigma, \varepsilon]\!]_\iota\} & \iota \in H \\
(\sigma, x := \varepsilon@\iota) & \Rightarrow_{\mathcal{A}} & \sigma\{x \mapsto [\![\mathit{rewrite}_{\mathcal{A}}(\sigma_C, \varepsilon)]\!]_\iota\} & \iota \in C
\end{array}$$

$$\begin{array}{rcll}
(\sigma, \mathtt{assert}(\varepsilon_1 = \varepsilon_2)@\iota) & \Rightarrow_{\mathcal{A}} & \sigma & \text{if } [\![\sigma, \varepsilon_1]\!]_\iota = [\![\sigma, \varepsilon_2]\!]_\iota \text{ or } \iota \in C \\
(\sigma, \mathtt{assert}(\phi(\varepsilon))@\iota) & \Rightarrow_{\mathcal{A}} & \bot & \text{if } \neg\phi(\sigma, [\![\sigma, \varepsilon]\!]_\iota)
\end{array}$$

$$(\sigma, x := \varepsilon@\iota) \Rightarrow \sigma\{x \mapsto [\![\sigma, \varepsilon]\!]_\iota\} \qquad \frac{(\sigma_1, \varepsilon_1) \Rightarrow \bot}{(\sigma_1, \varepsilon_1; \varepsilon_2) \Rightarrow \bot}$$

Value
$$\Gamma, \varnothing \vdash_\iota v : \varnothing$$

Secret
$$\Gamma, \varnothing \vdash_\iota \mathtt{s}[w] : \{\mathtt{s}[w]@\iota\}$$

Rando
$$\Gamma, \varnothing \vdash_\iota \mathtt{r}[w] : \{\mathtt{r}[w]@\iota\}$$

Mesg
$$\Gamma, \varnothing \vdash_\iota \mathtt{m}[w] : \Gamma(\mathtt{m}[w]@\iota)$$

PubM
$$\Gamma, \varnothing \vdash_\iota \mathtt{p}[w] : \Gamma(\mathtt{p}[w])$$

Share
$$\frac{\Gamma, R \vdash_\iota \varepsilon : T \qquad \oplus \in \{+, -\}}{\Gamma, R; \mathtt{r}[w]@\iota \vdash_\iota \varepsilon \oplus \mathtt{r}[w] : \{c(\mathtt{r}[w]@\iota, T)\}}$$

Binop
$$\frac{\Gamma, R_1 \vdash_\iota \varepsilon_1 : T_1 \qquad \Gamma, R_1 \vdash_\iota \varepsilon_1 : T_1 \qquad \oplus \in \{+, -, *\}}{\Gamma, R_1; R_2 \vdash_\iota \varepsilon_1 \oplus \varepsilon_2 : T_1 \cup T_2}$$

Command
$$\frac{\Gamma, R_1 \vdash_\iota \varepsilon : T \qquad \Gamma; x : T, R_2, E \cup \{x = \lfloor \varepsilon@\iota \rfloor\} \vdash \pi : \Gamma'}{\Gamma, R_1; R_2, E \vdash x := \varepsilon@\iota; \pi : \Gamma'}$$

Assert
$$\frac{E \vdash \lfloor \varepsilon_1@\iota \rfloor = \lfloor \varepsilon_2@\iota \rfloor \qquad \Gamma; R, E \vdash \pi : \Gamma'}{\Gamma, R, E \vdash \mathtt{assert}(\varepsilon_1 = \varepsilon_2)@\iota; \pi : \Gamma'}$$

Auth
$$\frac{E \vdash \lfloor \phi_{\text{auth}}(\mathtt{m}[w]) \rfloor \qquad \Gamma; \mathtt{m}[w]@\iota :\uparrow \Gamma(\mathtt{m}[w]@\iota), R, E \vdash \pi : \Gamma'}{\Gamma, R, E \vdash \mathtt{assert}(\phi_{\text{auth}}(\mathtt{m}[w]))@\iota; \pi : \Gamma'}$$

Term
$$\Gamma, \varnothing \vdash \varnothing : \Gamma, \varnothing$$

$$\ell \in \text{Field},\ y \in \text{EVar},\ f \in \text{FName}$$

$$\begin{array}{rcl}
e & ::= & v \mid \mathtt{r}[e] \mid \mathtt{s}[e] \mid \mathtt{m}[e] \mid \mathtt{p}[e] \mid e\ \mathit{binop}\ e \mid \mathtt{let}\ y = e\ \mathtt{in}\ e \mid \\
 & & f(e,\ldots,e) \mid \{\ell = e;\ldots;\ell = e\} \mid e.\ell \\
\mathbf{c} & ::= & \mathtt{m}[e]@e := e@e \mid \mathtt{p}[e] := e@e \mid \mathtt{out}@e := e@e \mid \mathtt{assert}(e = e)@e \mid \\
 & & f(e,\ldots,e) \mid \mathbf{c};\mathbf{c} \mid \mathtt{pre}(E) \mid \mathtt{post}(E) \\
\mathit{binop} & ::= & + \mid - \mid * \mid {+}{+} \\
v & ::= & w \mid \iota \mid \varepsilon \mid \{\ell = v;\ldots;\ell = v\} \\
\mathit{fn} & ::= & f(y,\ldots,y)\{e\} \mid f(y,\ldots,y)\{\mathbf{c}\} \\
\phi & ::= & \mathtt{r}[e]@e \mid \mathtt{s}[e]@e \mid \mathtt{m}[e]@e \mid \mathtt{p}[e] \mid \mathtt{out}@e \mid \phi + \phi \mid \phi - \phi \mid \phi * \phi \\
E & ::= & \phi = \phi \mid E \wedge E
\end{array}$$

$$\frac{e[v/y] \Rightarrow v'}{\mathtt{let}\ y = v\ \mathtt{in}\ e \Rightarrow v'}$$

$$\frac{C(f) = y_1,\ldots,y_n,\ e \qquad e_1 \Rightarrow v_1 \cdots e_n \Rightarrow v_n \qquad e[v_1/y_1]\cdots[v_n/y_n] \Rightarrow v}{f(e_1,\ldots,e_n) \Rightarrow v}$$

$$\frac{e_1 \Rightarrow v_1 \cdots e_n \Rightarrow v_n}{\{\ell_1 = e_1;\ldots;\ell_n = e_n\} \Rightarrow \{\ell_1 = v_1;\ldots;\ell_n = v_n\}} \qquad \frac{e \Rightarrow \{\ldots;\ell = v;\ldots\}}{e.\ell \Rightarrow v} \qquad \frac{e_1 \Rightarrow w_1 \qquad e_2 \Rightarrow w_2}{e_1{+}{+}e_2 \Rightarrow w_1 w_2}$$

$$\frac{e_1 \Rightarrow \varepsilon_1 \qquad e_2 \Rightarrow \varepsilon_2 \qquad e \Rightarrow \iota}{(\pi,(E_1,E_2),\mathsf{on},\mathtt{assert}(e_1 = e_2)@e) \Rightarrow (\pi;\mathtt{assert}(\varepsilon_1 = \varepsilon_2)@\iota,(E_1,E_2 \wedge \lfloor \varepsilon_1@\iota \rfloor = \lfloor \varepsilon_2@\iota \rfloor),\mathsf{on})}$$

$$\frac{e_1 \Rightarrow \varepsilon_1 \qquad e_2 \Rightarrow \varepsilon_2 \qquad e \Rightarrow \iota}{(\pi,(E_1,E_2),\mathtt{off},\mathtt{assert}(e_1 = e_2)@e) \Rightarrow (\pi;\mathtt{assert}(\varepsilon_1 = \varepsilon_2)@\iota,(E_1,E_2,\mathtt{off})}$$

$$\frac{e_1 \Rightarrow w \qquad e_2 \Rightarrow \iota_1 \qquad e_3 \Rightarrow \varepsilon \qquad e_4 \Rightarrow \iota_2}{(\pi,(E_1,E_2),\mathsf{on},\mathtt{m}[e_1]@e_2 := e_3@e_4) \Rightarrow (\pi;\mathtt{m}[w]@\iota_1 := \varepsilon@\iota_2,(E_1 \wedge \mathtt{m}[w]@\iota_1 = \lfloor \varepsilon@\iota_2 \rfloor, E_2),\mathsf{on})}$$

$$\frac{e_1 \Rightarrow w \qquad e_2 \Rightarrow \iota_1 \qquad e_3 \Rightarrow \varepsilon \qquad e_4 \Rightarrow \iota_2}{(\pi,(E_1,E_2),\mathtt{off},\mathtt{m}[e_1]@e_2 := e_3@e_4) \Rightarrow (\pi;\mathtt{m}[w]@\iota_1 := \varepsilon@\iota_2,(E_1,E_1),\mathtt{off})}$$

$$(\pi,(E_1,E_2),\mathsf{on},\mathtt{pre}(E)) \Rightarrow (\pi,E_1,E_2 \wedge E,\mathtt{off})$$

$$(\pi,(E_1,E_2),\mathtt{off},\mathtt{post}(E)) \Rightarrow (\pi,(E_1 \wedge E,E_2),\mathsf{on})$$

$$\frac{(\pi_1,(E_{11},E_{12}),sw_1,\mathbf{c}_1) \Rightarrow (\pi_2,(E_{21},E_{22}),sw_2) \qquad (\pi_2,(E_{21},E_{22}),sw_2,\mathbf{c}_2) \Rightarrow (\pi_3,(E_{31},E_{32}),sw_3)}{(\pi_1,(E_{11},E_{12}),sw_1,\mathbf{c}_1;\mathbf{c}_2) \Rightarrow (\pi_3,(E_{31},E_{32}),sw_3)}$$

$$\frac{\begin{array}{c} C(f) = y_1,\ldots,y_n,\ \mathbf{c} \\ e_1 \Rightarrow v_1 \cdots e_n \Rightarrow v_n \qquad (\pi_1,(E_{11},E_{12}),sw_1,\mathbf{c}[v_1/y_1,]\cdots[v_n/y_n]) \Rightarrow (\pi_2,(E_{21},E_{22}),sw_2) \end{array}}{(\pi_1,(E_{11},E_{12}),sw_1,f(e_1,\ldots,e_n)) \Rightarrow (\pi_2,(E_{21},E_{22}),sw_2)}$$

```
encodegmw(in, i1, i2) {
```

```
      m[in]@i2 := (s[in] xor r[in])@i2;
      m[in]@i1 := r[in]@i2
    }

    andtablegmw(b1, b2, r) {
      let r11 = r xor (b1 xor true) and (b2 xor true) in
      let r10 = r xor (b1 xor true) and (b2 xor false) in
      let r01 = r xor (b1 xor false) and (b2 xor true) in
      let r00 = r xor (bl xor false) and (b2 xor false) in
      { row1 = r11; row2 = r10; row3 = r01; row4 = r00 }
    }

    andgmw(z, x, y) {
      pre();
      let r = r[z] in
      let table = andtablegmw(m[x],m[y],r) in
      m[z]@2 := OT4(m[x],m[y],table,2,1);
      m[z]@1 := r@1;
      post(m[z]@1 xor m[z]@2 == (m[x]@1 xor m[x]@2) and (m[y]@1 xor m[y]@2))
    }

    xorgmw(z, x, y) {
      m[z]@1 := (m[x] xor m[y])@1; m[z]@2 := (m[x] xor m[y])@2;
    }

    decodegmw(z) {
      p["1"] := m[z]@1; p["2"] := m[z]@2;
      out@1 := (p["1"] xor p["2"])@1;
      out@2 := (p["1"] xor p["2"])@2
    }

    encodegmw("x",2,1);
    encodegmw("y",2,1);
    encodegmw("z",1,2);
    andgmw("g1","x","z");
    xorgmw("g2","g1","y");
    decodegmw("g2")
    pre();
    post(out@1 == (s["x"]@1 and s["z"]@2) xor s["y"]@1)


  secopen(w1,w2,w3,i1,i2) {
      pre(m[w1++"m"]@i2 == m[w1++"k"]@i1 + (m["delta"]@i1 * m[w1++"s"]@i2 /\
          m[w1++"m"]@i2 == m[w1++"k"]@i1 + (m["delta"]@i1 * m[w1++"s"]@i2));
      let locsum =  macsum(macshare(w1), macshare(w2)) in
      m[w3++"s"]@i1 := (locsum.share)@i2;
      m[w3++"m"]@i1 := (locsum.mac)@i2;
      auth(m[w3++"s"],m[w3++"m"],mack(w1) + mack(w2),i1);
```

```
    m[w3]@i1 := (m[w3++"s"] + (locsum.share))@i1
}


_open(x,i1,i2){
  m[x++"exts"]@i1 := m[x++"s"]@i2;
  m[x++"extm"]@i1 := m[x++"m"]@i2;
  assert(m[x++"extm"] == m[x++"k"] + (m["delta"] * m[x++"exts"]));
  m[x]@i1 := (m[x++"exts"] + m[x++"s"])@i2
}`

_sum(z, x, y,i1,i2) {
    pre(m[x++"m"]@i2 == m[x++"k"]@i1 + (m["delta"]@i1 * m[x++"s"]@i2 /\
        m[y++"m"]@i2 == m[y++"k"]@i1 + (m["delta"]@i1 * m[y++"s"]@i2));
    m[z++"s"]@i2 := (m[x++"s"] + m[y++"s"])@i2;
    m[z++"m"]@i2 := (m[x++"m"] + m[y++"m"])@i2;
    m[z++"k"]@i1 := (m[x++"k"] + m[y++"k"])@i1;
    post(m[z++"m"]@i2 == m[z++"k"]@i1 + (m["delta"]@i1 * m[z++"s"]@i2)
}

sum(z,x,y) { _sum(z,x,y,1,2);_sum(z,x,y,2,1) }

open(x) { _open(x,1,2); _open(x,2,1) }


sum("a","x","d");
open("d");
sum("b","y","e");
open("e");
let xys =
    macsum(macctimes(macshare("b"), m["d"]),
           macsum(macctimes(macshare("a"), m["e"]),
                  macshare("c")))
let xyk = mack("b") * m["d"] + mack("a") * m["e"] + mack("c")

secopen("a","x","d",1,2);
  secopen("a","x","d",2,1);
  secopen("b","y","e",1,2);
  secopen("b","y","e",2,1);
  let xys =
    macsum(macctimes(macshare("b"), m["d"]),
           macsum(macctimes(macshare("a"), m["e"]),
                  macshare("c")))
  in
  let xyk = mack("b") * m["d"] + mack("d") * m["d"] + mack("c")
  in
  secreveal(xys,xyk,"1",1,2);
  secreveal(maccsum(xys,m["d"] * m["e"]),
```

```
            xyk - m["d"] * m["e"],
            "2",2,1);
out@1 := (p[1] + p[2])@1;
out@2 := (p[1] + p[2])@2;
```