1 *Overture* SYNTAX AND SEMANTICS

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

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

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

$$(\sigma, x := \varepsilon\texttt{@}\iota) \Rightarrow \sigma\{x \mapsto [\![\sigma, \varepsilon]\!]_\iota\} \qquad \frac{(\sigma_1, \pi_1) \Rightarrow \sigma_2 \qquad (\sigma_2, \pi_2) \Rightarrow \sigma_3}{(\sigma_1, \pi_1; \pi_2) \Rightarrow \sigma_3}$$

2 *Overture* ADVERSARIAL SEMANTICS

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

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

$$\frac{(\sigma_1, \pi_1) \Rightarrow_{\mathcal{A}} \sigma_2 \qquad (\sigma_2, \pi_2) \Rightarrow_{\mathcal{A}} \sigma_3}{(\sigma_1, \pi_1; \pi_2) \Rightarrow_{\mathcal{A}} \sigma_3} \qquad \frac{(\sigma_1, \pi_1) \Rightarrow_{\mathcal{A}} \bot}{(\sigma_1, \pi_1; \pi_2) \Rightarrow_{\mathcal{A}} \bot}$$

$$\frac{(\sigma_1, \pi_1) \Rightarrow_{\mathcal{A}} \sigma_2 \qquad (\sigma_2, \pi_2) \Rightarrow_{\mathcal{A}} \bot}{(\sigma_1, \pi_1; \pi_2) \Rightarrow_{\mathcal{A}} \bot}$$

3 *Overture* CONSTRAINT TYPING

Value
$$\Gamma, \varnothing, E \vdash_\iota v : \varnothing \cdot \mathrm{High}$$

Secret
$$\Gamma, \varnothing, E \vdash_\iota \texttt{s[}w\texttt{]} : \{\texttt{s[}w\texttt{]@}\iota\} \cdot \mathcal{L}(\iota)$$

Rando
$$\Gamma, \varnothing, E \vdash_\iota \texttt{r[}w\texttt{]} : \{\texttt{r[}w\texttt{]@}\iota\} \cdot \mathcal{L}(\iota)$$

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

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

IntegrityWeaken
$$\frac{\Gamma, R, E \vdash_\iota \varepsilon : T \cdot \varsigma_1 \qquad \varsigma_1 \preceq \varsigma_2}{\Gamma, R, E \vdash_\iota \varepsilon : T \cdot \varsigma_2}$$

RandoDeduce
$$\frac{\Gamma, \varnothing, E \vdash_\iota \varepsilon : T \cdot \varsigma \qquad E \models \lfloor \varepsilon @ \iota \rfloor = \texttt{r[}w\texttt{]@}\iota'}{\Gamma, \varnothing, E \vdash_\iota \varepsilon : \{\texttt{r[}w\texttt{]@}\iota\} \cdot \varsigma}$$

Encode
$$\frac{\Gamma, R_1, E \vdash_\iota \varepsilon_1 : T \cdot \varsigma \qquad \Gamma, R_2, E \vdash_\iota \varepsilon_2 : \{\texttt{r[}w\texttt{]@}\iota\} \cdot \varsigma \qquad \oplus \in \{+, -\}}{\Gamma, R1; R_2; \texttt{r[}w\texttt{]@}\iota, E \vdash_\iota \varepsilon_1 \oplus \varepsilon_2 : \{c(\texttt{r[}w\texttt{]@}\iota, T)\} \cdot \varsigma}$$

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

Send
$$\frac{\Gamma, R, E \vdash_\iota \varepsilon : T \cdot \mathcal{L}(\iota) \qquad E' \models E \wedge x = \lfloor \varepsilon @ \iota \rfloor}{\Gamma, R, E \vdash x := \varepsilon @ \iota : \Gamma; x : T \cdot \mathcal{L}(\iota), E'}$$

Assert
$$\frac{E \models \lfloor \varepsilon_1 @ \iota \rfloor = \lfloor \varepsilon_2 @ \iota \rfloor}{\Gamma, R, E \vdash \texttt{assert(}\varepsilon_1 = \varepsilon_2\texttt{)@}\iota : \Gamma, E}$$

Seq
$$\frac{\Gamma_1, R_1, E_1 \vdash \pi_1 : \Gamma_2, E_2 \qquad \Gamma_2, R_2, E_2 \vdash \pi_2 : \Gamma_3, E_3}{\Gamma_1, R_1; R_2, E_1 \vdash \pi_1; \pi_2 : \Gamma_3, E_3}$$

Constraint
$$\frac{\Gamma_1, R, E_1 \vdash \pi : \Gamma_2, E_2 \qquad E_1' \models E_1' \qquad E_2 \models E_2'}{\Gamma_1, R, E_1' \vdash \pi : \Gamma_2, E_2'}$$

MAC
$$\frac{E \models \texttt{m[}wm\texttt{]@}\iota = \texttt{m[}wk\texttt{]@}\iota + (\texttt{m[delta]@}\iota * \texttt{m[}ws\texttt{]@}\iota) \qquad \Gamma(\texttt{m[}ws\texttt{]@}\iota) = T \cdot \varsigma}{\Gamma, R, E \vdash \texttt{assert(m[}wm\texttt{] = m[}wk\texttt{] + (m[delta] * m[}ws\texttt{]))@}\iota : \Gamma; \texttt{m[}ws\texttt{]@}\iota : T \cdot \mathrm{High}, E}$$

4 *Prelude* SYNTAX AND SEMANTICS

$$\begin{array}{rcl}
\multicolumn{3}{l}{\ell \in \mathrm{Field},\ y \in \mathrm{EVar},\ f \in \mathrm{FName}} \\
e & ::= & v \mid \texttt{r[}e\texttt{]} \mid \texttt{s[}e\texttt{]} \mid \texttt{m[}e\texttt{]} \mid \texttt{p[}e\texttt{]} \mid e\ \mathit{binop}\ e \mid \texttt{let}\ y = e\ \texttt{in}\ e \mid \\
& & f(e, \ldots, e) \mid \{\ell = e; \ldots; \ell = e\} \mid e.\ell \\
\mathbf{c} & ::= & \texttt{m[}e\texttt{]@}e := e\texttt{@}e \mid \texttt{p[}e\texttt{]} := e\texttt{@}e \mid \texttt{out@}e := e\texttt{@}e \mid \texttt{assert(}e = e\texttt{)@}e \mid \\
& & f(e, \ldots, e) \mid \mathbf{c}; \mathbf{c} \mid \texttt{pre(}E\texttt{)} \mid \texttt{post(}E\texttt{)} \\
\mathit{binop} & ::= & + \mid - \mid * \mid \texttt{++} \\
v & ::= & w \mid \iota \mid \varepsilon \mid \{\ell = v; \ldots; \ell = v\} \\
\mathit{fn} & ::= & f(y, \ldots, y)\{e\} \mid f(y, \ldots, y)\{\mathbf{c}\} \\
\phi & ::= & \texttt{r[}e\texttt{]@}e \mid \texttt{s[}e\texttt{]@}e \mid \texttt{m[}e\texttt{]@}e \mid \texttt{p[}e\texttt{]} \mid \texttt{out@}e \mid \phi + \phi \mid \phi - \phi \mid \phi * \phi \\
E & ::= & \phi = \phi \mid E \wedge E
\end{array}$$

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

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

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

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

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

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

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

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

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

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

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

5 EXAMPLES

```
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;
```