From $BitML^x$ to $BitML^{||}$

linen

IMDEA Software Institute

January 14, 2022

1 Description of $BitML^x$

$BitML^x$, an extension of BitML, is a simple domain-specific language, which allows us to specify smart contracts that regulate cryptocurrency exchanges among participants, who own coins in different bitcoin-based blockchains.

We compile $BitML^x$ to $BitML^{||}$.

2 Full Grammar of $BitML^x$

$$Program ::= \{G\}C$$

$$
\begin{aligned}
C &::= D \mid C \triangleright C \\
D &::= \\
&\quad put\ \vec{x}\ \&\ reveal\ \vec{a}\ if\ p.C \\
&\mid withdraw\ A \\
&\mid split\ \left(\vec{u}^B, \vec{u}^D\right) \rightarrow \vec{C} \\
&\mid A\ :\ D
\end{aligned}
\qquad
\begin{aligned}
G &::= \\
&\quad A\ :?\ \left(u^B, u^D\right)\ @\ \left(x^B, x^D\right) \\
&\mid A\ :!\ \left(\left(u^B, u^D\right)\ @\ \left(x^B, x^D\right)\right. \\
&\mid A\ :\ secret\ a \\
&\mid G \mid G
\end{aligned}
$$

$$
\begin{array}{ll}
p ::= & E ::= \\
\quad true & \quad N \\
\quad \mid\ p \wedge p & \quad \mid\ |a| \\
\quad \mid\ \neg\ p & \quad \mid\ E + E \\
\quad \mid\ E = E & \quad \mid\ E - E \\
\quad \mid\ E < E &
\end{array}
$$

3 Auxiliary functions

- $depC^B : PartG^B \rightarrow \{u^B_{col}\}$
- $dep^B : PartG^B \rightarrow \{u'^B\}$
- $depC^D : PartG^D \rightarrow \{u^D_{col}\}$
- $dep^D : PartG^D \rightarrow \{u'^D\}$

4 Compiler

$$
\text{C-Adv}\quad
\frac{
\begin{array}{l}
G = \Big(\|_{i\in I} A_i :?(u_i^B, u_i^D)@(x_i, y_i)\Big) \,|\, \Big(\|_{i\in J} B_i :!(u_i'^B, u_i'^D)@(x_i', y_i')\Big) \,|\, \Big(\|_{i\in K} C_i : secret\ a_i\Big) \\
C = D_1 \triangleright D_2 \ldots \triangleright D_m \qquad PartG \qquad |PartG| = n \\
u_B = \sum_{i\in J} u_i'^B \qquad u_D = \sum_{i\in J} u_i'^D \\
G^B = \Big(\|_{i\in I} A_i \ \ :?\ u_i^B\ @\ x_i\Big) \,|\, \Big(\|_{i\in J} B_i \ \ :!\ u_i'^B\ @\ x_i'\Big) \,|\, \Big(\|_{i\in K} C_i : secret\ a_i\Big) \\
\qquad\quad |\Big(\|_{i\in PartG} P_i \ \ :!\ u_{i\ col}^B\ @\ x_{i\ col}\Big) \,|\, \Big(\|_{i\in PartG\, j\in(1..m-1)} P_i : secret\ s_{ij}\Big) \\
G^D = \Big(\|_{i\in I} A_i \ \ :?\ u_i^D\ @\ y_i\Big) \,|\, \Big(\|_{i\in J} B_i \ \ :!\ u_i'^D\ @\ y_i'\Big) \,|\, \Big(\|_{i\in K} C_i : secret\ a_i\Big) \\
\qquad\quad |\Big(\|_{i\in PartG} P_i \ \ :!\ u_{i\ col}^D\ @\ y_{i\ col}\Big) \,|\, \Big(\|_{i\in PartG\, j\in(1..m-1)} P_i : secret\ \hat{s}_{ij}\Big) \\
u_{i\ col}^B = \sum_{j\in J} (n-1)\cdot u_j^B \quad \forall i \in PartG \qquad u_{col}^B = n\cdot(n-1)\cdot u_B \\
u_{i\ col}^D = \sum_{j\in J} (n-1)\cdot u_j^D \quad \forall i \in PartG \qquad u_{col}^D = n\cdot(n-1)\cdot u_D \\
C^B = \mathcal{B}_D^x\Big(C, P, u^B, u_{col}^B, \vec{s}, \vec{\hat{s}}, 1, 1, \vec{x}', \vec{x}^{col}\Big) \\
C^D = \mathcal{B}_D^x\Big(C, P, u^D, u_{col}^D, \vec{s}, \vec{\hat{s}}, 1, 1, \vec{y}', \vec{y}^{col}\Big) \\
\vec{s} = s_{11}\ldots s_{1m} s_{21}\ldots s_{2m}\ldots s_{n1}\ldots s_{nm} \\
\vec{\hat{s}} = \hat{s}_{11}\ldots \hat{s}_{1m} \hat{s}_{21}\ldots \hat{s}_{2m}\ldots \hat{s}_{n1}\ldots \hat{s}_{nm} \\
\vec{x}^{col},\ \vec{y}^{col},\ \vec{s},\ \vec{\hat{s}} \qquad fresh
\end{array}
}{
\mathcal{B}_{adv}^x(\{G\}C) = \{G^B\}C^B, \{G^D\}C^D
}
$$

$$
\text{C-With}\quad
\frac{
\begin{array}{l}
D = withdraw\ A \triangleright \mathcal{D} \qquad |P| = n \\
D' = split\ u \to withdraw\ A \\
\qquad\qquad |\ u_{col}\backslash n \to withdraw P_1 \\
\qquad\qquad |\ldots \\
\qquad\qquad |\ u_{col}\backslash n \to withdraw P_n \\
D'' = \mathcal{B}_A\Big(P, u, u_{col}, \vec{s}, \vec{\hat{s}}, \vec{x}, \vec{x}^{col}, i\Big) \\
D''' = \mathcal{B}_D^x\Big(\mathcal{D}, P, u, u_{col}, \vec{s}, \vec{\hat{s}}, (i+1), \big(i\cdot t + T_{cheat}\big), x', x^{col}\Big)
\end{array}
}{
\begin{array}{l}
\mathcal{B}_D^x\Big(D, P, u, u_{col}, \vec{s}, \vec{\hat{s}}, i, t, x', x^{col}\Big) = \\
\quad reveal\ s_{i1} \vee reveal\ s_{i2} \vee \ldots \vee reveal\ s_{in}.\ D' \\
\quad +\ after\ i\cdot t\ :\ D'' \\
\quad +\ after\ \big(i\cdot t + T_{cheat}\big)\ :\ D'''
\end{array}
}
$$

$$\text{C-WithSplit}\quad \frac{dep^B(P_i, \vec{x}) = u_i}{\begin{aligned}&\mathcal{B}_A\Big(P, u, u_{col}, \vec{s}, \vec{\hat{s}}, \vec{x}, \vec{x}^{col}, i\Big) = \\ &\quad reveal\ \hat{s}_{i1}.\ split\ \big(u_2 + u_{col} \backslash (n-1)\big) \rightarrow withdraw\ P_2 \\ &\qquad\qquad\quad |\ldots \\ &\qquad\qquad\quad |\quad \big(u_n + u_{col} \backslash (n-1)\big) \rightarrow withdraw\ P_n \\ &\quad \vdots \\ &+ reveal\ \hat{s}_{in}.\ split\ \big(u_1 + u_{col} \backslash (n-1)\big) \rightarrow withdraw\ P_1 \\ &\qquad\qquad\quad |\ldots \\ &\qquad\qquad\quad |\quad \big(u_{n-1} + u_{col} \backslash (n-1)\big) \rightarrow withdraw\ P_{n-1}\end{aligned}}$$

$$\text{C-RevAuth}\quad \frac{\begin{aligned}&D \equiv A_1 :\ \ldots\ A_n : put\ \vec{z}\ \&\ reveal\ \vec{a}\ if\ p.\mathcal{D}_1 \triangleright \mathcal{D}_2 \\ &\mathcal{D}_1 \neq A_1 :\ \ldots\ A_n : put\ \vec{z}\ \&\ reveal\ \vec{a}\ if\ p.\mathcal{C}_1 \triangleright \mathcal{C}_2 \\ &\mathcal{D} = \mathcal{D}_1 \triangleright \mathcal{D}_2 \\ &\mathcal{D}_1' = reveal\ s_{i1} \vee \ldots \vee reveal\ s_{in}\ . \\ &\qquad A_1 :\ \ldots\ A_n : put\ \vec{z}\ \&\ reveal\ \vec{a}\ if\ p. \\ &\qquad \mathcal{B}_C\Big(\mathcal{D}, P, u, u_{col}, \vec{s}, \vec{\hat{s}}, i, t, \vec{x}', \vec{x}^{col}\Big)\end{aligned}}{\mathcal{B}_D^x\Big(D, P, u, u_{col}, \vec{s}, \vec{\hat{s}}, i, t, \vec{x}', \vec{x}^{col}\Big) = \mathcal{D}_1'}$$