XVI International Workshop on Rewriting Logic and its Applications

A Theory of Composable Lingos for Protocol Dialects

Víctor García[1] Santiago Escobar[1] Catherine Meadows Jose Meseguer[2]

[1]Universidad Politécnica de Valéncia, Spain

[2]University of Illinois at Urbana-Champaign, USA

What is MTD?
MTD stands for Moving Target Defense. There are many forms of MTD: watermarking, signal jamming, protocol dialects.

What is a Dialect?
It is a form of *formal pattern*, which are methods for formally specifying generic solutions to distributed systems problems. In our case, the **problem** is security for enclave members, the **solution** are Protocol Dialects.

Everything presented next is an improvement and an extension from what was published in ESORICS'23.

What is a Lingo?

A lingo Λ is a data transformation between data types D_1 and D_2, with a one-sided inverse. The transformation is parametric on a parameter value a belonging to a parameter set A. For each parameter a, data from D_1 is transformed into data of D_2, which, using the same parameter a, can be transformed back into the original data from D_1.

Definition 1 (Lingo)

We can formalize a *lingo* Λ as a 5-tuple $\Lambda = (D_1, D_2, A, f, g)$, where

$$
\begin{aligned}
&D_1, D_2 \text{ and } A \text{ are non-empty sets,}\\
&f : D_1 \times A \to D_2,\\
&g : D_2 \times A \to D_1\\
&\text{such that } \forall d_1 \in D_1, \forall a \in A, g(f(d_1, a), a) = d_1
\end{aligned}
$$

Example 2 (Lingo Λ_{xor})

Let $\Lambda_{xor} = (\mathbb{N}, \mathbb{N}, \mathbb{N}, \oplus, \oplus)$, with $\oplus : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ the bitwise exclusive or operation.

Let $d_1, d_2, a \in \mathbb{N}$, then by substitution, $d_2 = f(d_1, a) = (d_1 \oplus a)$, so it follows that $g(f(d_1, a), a) = ((d_1 \oplus a) \oplus a) = ((a \oplus a) \oplus d_1) = (0 \oplus d_1) = d_1$, as stated by (i) Definition 1 and (ii) the equational properties given by the XOR theory.

What are the Desirable Properties of a Lingo?

The purpose of a lingo $\Lambda = (D_1, D_2, A, f, g)$, is to obfuscate the communication between a sender, *Alice*, and a receiver, *Bob*, who use a common, dynamically changing, **secret parameter** $a \in A$. To make it hard for an intruder to interfere with their communication it would be desirable that, for example,

1. The probability that the attacker can guess the secret parameter a before it is used is low.
2. Non-compliant messages from an attacker, i.e., messages that do not use the current secret parameter a, can be detected and discarded (Λ is f-**checkable**).
3. The attacker cannot modify a legitimate message to make it compliant with the current secret parameter a (Λ is not **malleable**).
4. It is hard for an attacker to impersonate a legitimate protocol participant.

Outline

Definition 3 (f-Checkable Lingo)

A lingo $\Lambda = (D_1, D_2, A, f, g)$ is called f-checkable iff $\forall a \in A, \exists d_2 \in D_2$ such that $\nexists d_1 \in D_1$ such that $d_2 = f(d_1, a)$.

In natural language: there is junk in D_2 that f cannot produce so the Dialect can detect it as malicious or as an anomaly to be discarded.

Example 4 (Lingo D&C)

$\Lambda_{d\&c} = (\mathbb{N}, \mathbb{N} \times \mathbb{N}, \mathbb{N}, f, g)$, with:

- $f(n, a) = (quot(n + (a + 2), a + 2), rem(n + (a + 2), a + 2))$
- $g((x, y), a) = (x \cdot (a + 2)) + y - (a + 2)$

There is an automatic transformation $\Lambda \mapsto \Lambda^\sharp$ transforming any lingo Λ into an f-checkable lingo.

Theorem 5

Let $\Lambda = (D_1, D_2, A, f, g)$ be a lingo, where we assume that D_1, D_2 and A are computable data-types and $|D_1| \geq 2$. Then, $\Lambda^\sharp = (D_1, D_2 \times D_2, A \otimes A, f^\sharp, g^\sharp)$ is an f-checkable lingo, where:

- $A \otimes A = A \times A \setminus id_A$, *with* $id_A = \{(a, a) \in A^2 | a \in A\}$
- $f^\sharp(d_1, (a, a')) = (f(d_1, a), f(d_1, a'))$
- $g^\sharp((d_2, d_2'), (a, a')) = g(d_2, a)$.

Example 6 (f-checkable transformed version of Λ_{xor})

The lingo transformation $\Lambda \mapsto \Lambda^\sharp$ maps Λ_{xor} in Example 2 to the f-checkable lingo $\Lambda^\sharp_{xor} = (\mathbb{N}, \mathbb{N} \times \mathbb{N}, \mathbb{N} \otimes \mathbb{N}, \oplus^\sharp, \oplus^\sharp)$.

Outline

Malleable Lingos

Call a lingo Λ *malleable* if an intruder can disrupt the communication between an honest sender *Alice* and an honest receiver *Bob* in a protocol $\mathcal{P}$ whose messages are obfuscated with lingo Λ by producing a *compliant* message supposedly sent from *Alice* to *Bob* using a secret parameter $a \in A$ of Λ, but actually sent by the intruder.

Definition 7 (Malleable Lingo)

Λ is called *malleable* iff there exists a Σ_Λ-term $t(x, y)$ (recipe) of sort D_2 with free variables x, y of respective sorts D_2 and A, as well as a subset $A_0 \subseteq A$, such that $\forall d_1 \in D_1,\ \forall a \in A,\ \forall a' \in A_0$,

1. $f(d_1, a) \neq t(f(d_1, a), a')$
2. $\exists d_1' \in D_1$ such that $t(f(d_1, a), a') = f(d_1', a)$.

Examples of Malleable Lingos

Example 8 (Lingo Λ_{xor})

Let $\Lambda_{xor} = (\mathbb{N}, \mathbb{N}, \mathbb{N}, \oplus, \oplus)$, be the lingo from Example 2.

Λ_{xor} is a malleable lingo. The recipe $t(x, y)$ is $x \oplus y$, because $\forall d_1, d_1', a, a' \in \mathbb{N}, ((d_1 \oplus a) \oplus a') = (d_1' \oplus a)$, where $d_1' = d_1 \oplus a'$.

There are f-checkable lingos that are malleable.

Example 9 (Lingo XOR#)

The f-checkable lingo $\Lambda_{xor}^{\sharp}$ obtained from Λ_{xor} by applying the $\Lambda \mapsto \Lambda^{\sharp}$ transformation is malleable. with recipe $t(x, y) = [p_1(x) \oplus p_1(y), p_2(x) \oplus p_1(y)]$.

Horizontal Composition

The horizontal composition of two lingos with the same D_1 is a lingo that behaves as the first lingo or as the second one depending on the result of the pseudo-random throw of a biased coin. The coin bias β should be chosen so as to favor the lingo that is deemed stronger.

Definition 10 (Horizontal Composition)

Let $\overrightarrow{\Lambda}$ be a finite family of lingos $\overrightarrow{\Lambda} = \{\Lambda_i\}_{1 \leq i \leq k}$, $k \geq 2$, all having the same input data type D_1, i.e., $\Lambda_i = (D_1, D_{2.i}, A_i, f_i, g_i)$, $1 \leq i \leq k$. Let $\overrightarrow{d}_0 = (d_{0.1}, \ldots, d_{0.k})$ be a choice of default $D_{2.i}$-values, $d_{0.i} \in D_{2.i}$, $1 \leq i \leq k$. Then, the *horizontal composition* of the lingos $\overrightarrow{\Lambda}$ with default $D_{2.i}$-values $\overrightarrow{d}_0$ is the lingo:

$$\bigoplus_{\overrightarrow{d_0}} \overrightarrow{\Lambda} = (D_1, \bigcup_{1 \leq i \leq k} D_{2.i}, \bigcup_{1 \leq i \leq k} A_i \times \{i\}, \oplus \overrightarrow{f}, \oplus \overrightarrow{g})$$

where, for each $d_1 \in D_1$, $d_2 \in \bigcup_{1 \leq i \leq k} D_{2.i}$, and $a_i \in A_i$, $1 \leq i \leq k$,

1. $\oplus \overrightarrow{f} = \lambda(d_1, (a_i, i)).f_i(d_1, a_i)$,
2. $\oplus \overrightarrow{g} = \lambda(d_2, (a_i, i))$. **if** $d_2 \in D_{2.i}$ **then** $g_i(d_i, a_i)$ **else** $g_i(d_{0.i}, a_i)$.

$\bigoplus_{\overrightarrow{d_0}} \overrightarrow{\Lambda}$ is indeed a lingo: $\oplus \overrightarrow{g}(\oplus \overrightarrow{f}(d_1, (a_i, i)), (a_i, i)) = g_i(f_i(d_1, a_i), a_i) = d_1$.

Example of Horizontal Composition

Example 11 (Horizontal composition of XOR and D&C)

Let Λ_{xor} be the exclusive or lingo from Example 8. Let $\Lambda_{D\&C}$ be the divide and check lingo from Example 4. Then, the horizontal composition of Λ_{xor} and $\Lambda_{D\&C}$ is the lingo $\Lambda_{xor} \oplus_\beta \Lambda_{D\&C} = (\mathbb{N}, \mathbb{N} \cup (\mathbb{N} \times \mathbb{N}), \mathbb{N} \uplus \mathbb{N}, \oplus \oplus_\beta \mathit{divide}, \oplus \oplus_\beta \mathit{check})$.

Functional Composition

The functional composition $\Lambda \odot \Lambda'$ of lingos Λ and Λ', first alters a message with Λ, and then further alters the result with Λ'. This lingo composition can greatly increase the choice of parameters and can make life much harder for attackers.

Definition 12 (Functional Composition)

Given lingos $\Lambda = (D_1, D_2, A, f, g)$ and $\Lambda' = (D_2, D_3, A', f', g')$, their *functional composition* is the lingo $\Lambda \odot \Lambda' = (D_1, D_3, A \times A', f \cdot f', g * g')$, where for each $d_1 \in D_1$, $d_3 \in D_3$, and $(a, a') \in A \times A'$,

- $f \cdot f'(d_1, (a, a')) =_{def} f'(f(d_1, a), a')$,
- $g * g'(d_3, (a, a')) =_{def} g(g'(d_3, a'), a)$.

$(D_1, D_3, A \times A', f \cdot f', g * g')$ is indeed a lingo, since we have:

$$\begin{aligned} g * g'(f \cdot f'(d_1, (a, a')), (a, a')) &= g(g'(f'(f(d_1, a), a'), a'), a) \\ &= g(f(d_1, a), a) = d_1. \end{aligned}$$

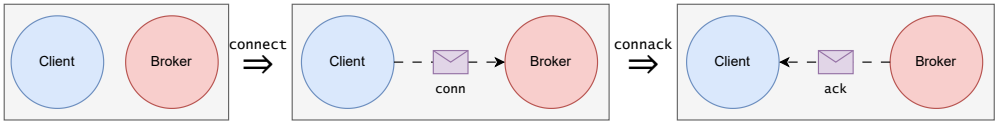
Example 13 (Functional Composition of XOR and D&C)

Consider lingos Λ_{xor} and $\Lambda_{d\&c}$, given in Examples 2 and 4 respectively. We can functionally composed them as $\Lambda_{xor \odot d\&c}$ where for each $d_1 \in \mathbb{N}$, $d_3 \in \mathbb{N} \times \mathbb{N}$, and $(a, a') \in \mathbb{N} \times \mathbb{N}$,

- $xor \cdot divide(d_1, (a, a')) = divide(xor(d_1, a), a')$
- $xor * check(d_3, (a, a')) = xor(check(d_3, a'), a)$

Protocols as Rewrite Theories

A protocol $\mathcal{P}$ is modeled as a collection of objects communicating by message passing. Message send and receive actions are specified by rewrite rules.

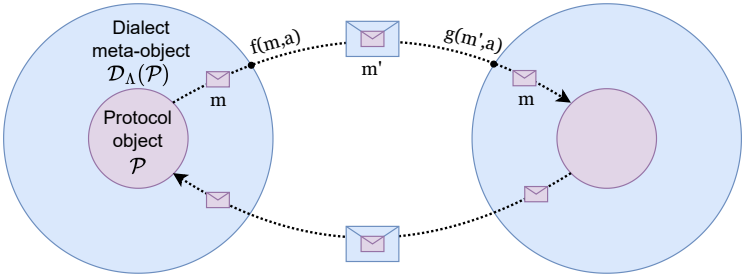



```
rl [connect] :
    < C : Client | brk : B, Atts >
 => < C : Client | brk : B, Atts >
    (from C to B : conn) .
```

```
rl [connack] :
    < B : Broker | peer : none, Atts >
    (from C to B : conn)
 => < B : Broker | peer : C, Atts >
    (from B to C : ack) .
```

What is a Dialect?

Dialects are protocol transformations of the form $\mathcal{P}(D) \mapsto \mathcal{D}(\Lambda(D), \mathcal{P}(D))$, where $\mathcal{D}(\Lambda(D), \mathcal{P}(D))$, is a protocol that wraps each object of $\mathcal{P}(D)$ inside a *dialect meta-actor* that uses lingo $\Lambda(D)$ to obfuscate the communication between the honest protocol actors of $\mathcal{P}(D)$.



What is a Dialect?

Meta-Object actions are specified by rewrite rules, and are both *protocol* and lingo-generic. Thus, the meta-object rules of a Dialect can be applied to a wide range of underlying protocols, including to protocols transformed by other dialects. The rule below shows how the meta-object *obfuscates* a message before being sent.

$$
\begin{aligned}
\texttt{rl}\ [\texttt{out}]\ : &< O_1 : DC \mid \texttt{conf} : (< O_1 : C \mid atts > \ (from\ O_1\ to\ O_2 : P) \cup\ M), \\
&\qquad \texttt{peer-counters} : R,\ atts' > \\
\texttt{=>}\ &< O_1 : DC \mid \texttt{conf} : (< O_1 : C \mid atts > \ M \setminus (from\ O_1\ to\ O_2 : P)), \\
&\qquad \texttt{update}(\texttt{peer-counters} : R, atts') > \\
&(from\ O_1\ to\ O_2 : f(P, param(R[O_2]))) \ .
\end{aligned}
$$

Note that *the underlying protocol is not changed at all*, making dialects highly modular.

Conclusions

- Refine the definition of Lingo and Dialects.
- Provide lingo transformations to obtain stronger lingos.
- Propose two new lingo composition operations: Horizontal and Functional.
- Everything presented here is not just theoretical but executable in Maude.

Future Work

- Probabilistic Model Checking to study the strength of lingos and their compositions against an *on-path* attacker.
- New kinds of lingos: authenticating lingos, data adaptors.
- New composition operations: tupling, product of lingos.
- *Faster* moving targets: Dialects based on time (Periodic).