A Review of: Protocol Dialects as Formal Patterns

Daniel Galán[1], Víctor García[2], Santiago Escobar[2],
Catherine Meadows[3], Jose Meseguer[4]

[1]ETH Zurich, Switzerland
[2]VRAIN, Universitat Politècnica de València, Spain
[3]Naval Research Laboratory, USA
[4]University of Illinois at Urbana-Champaign, USA

Introduction

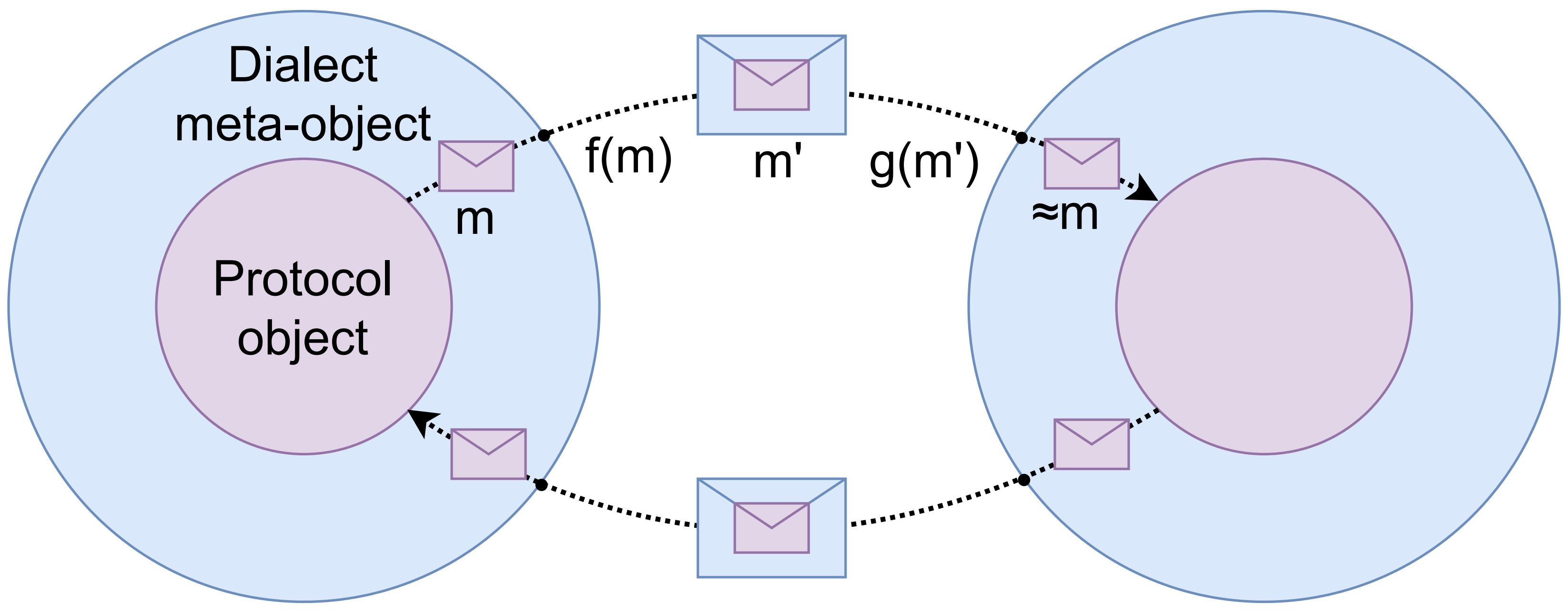
- *Protocol Dialects* are modifications to existing protocols (by means of *lingos*) that provide lightweight security.
- Network of mutually trusting principals, e.g. an enterprise network, or an IoT network with reduced computing capabilities.
- We present a framework based on *Formal Patterns* that allow us to generalize the notion of dialect with two important consequences: dialects become composable, and composed dialects can be harder to break.

Dialects as Formal Patterns

Dialects are made generic on two dimensions:

i) ***protocol generic,*** allowing the dialect to work on a broad class of protocols, and

ii) ***lingo generic,*** allowing the dialect to use different lingos (even simultaneously) to achieve higher levels of obfuscation.



Lingos

Lingos are invertible message transformations (called $\boldsymbol{f}$ and $\boldsymbol{g}$) used to obfuscate messages $\vec{\boldsymbol{m}}$. These transformations are parametric, and theory-generic. This means, our lingos can work on different security levels and on different protocols. Note that $\boldsymbol{f}$ and $\boldsymbol{g}$ are parametrically inverse functions, i.e.,

$$g(f(\vec{m}, a), a) = \vec{m}$$

Dialects

Dialects are protocol transformations of the form

$$D \to D(P, \Lambda_1(P), \ldots, \Lambda_n(P))$$

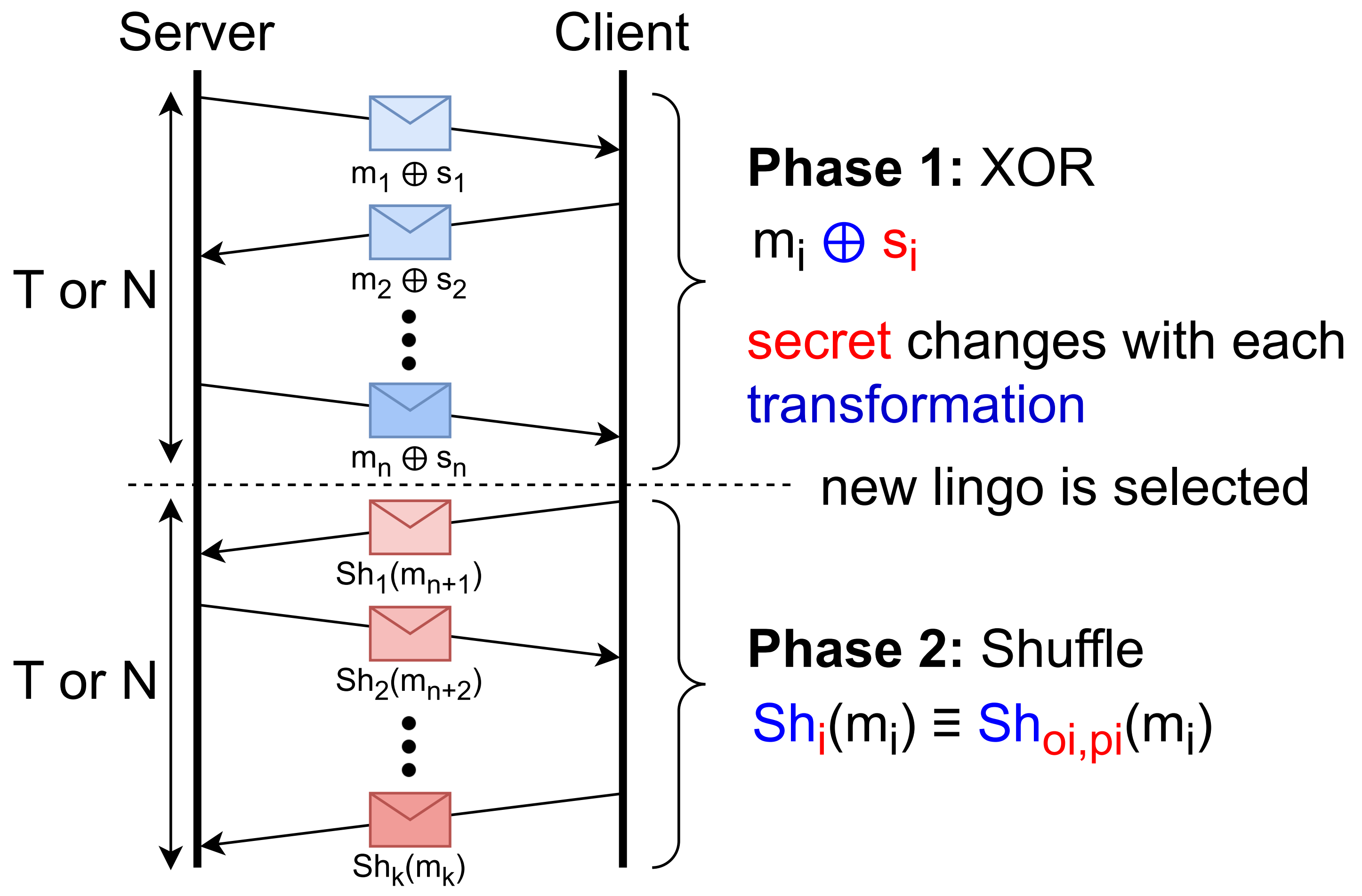
where the dialect endows protocol $\boldsymbol{P}$ with $\boldsymbol{n}$ Lingos $\vec{\boldsymbol{\Lambda}}_{\boldsymbol{n}}$, and wraps each protocol participant inside a dialect meta-object that uses lingos in $\vec{\boldsymbol{\Lambda}}_{\boldsymbol{n}}$ to obfuscate the communication.

Dialects as Moving Targets

Every dialect should be a moving target, but some dialects move *faster* than others. We have the following two classes of dialects:

i) ***Static* dialects,** where the lingo of the dialect never changes, but the secret parameter does.

ii) ***Dynamic* dialects**, where the lingo, and the secret parameter, change dynamically (**periodic** or **aperiodic**).



Phase 1: XOR

$m_i \oplus s_i$

secret changes with each transformation

new lingo is selected

Phase 2: Shuffle

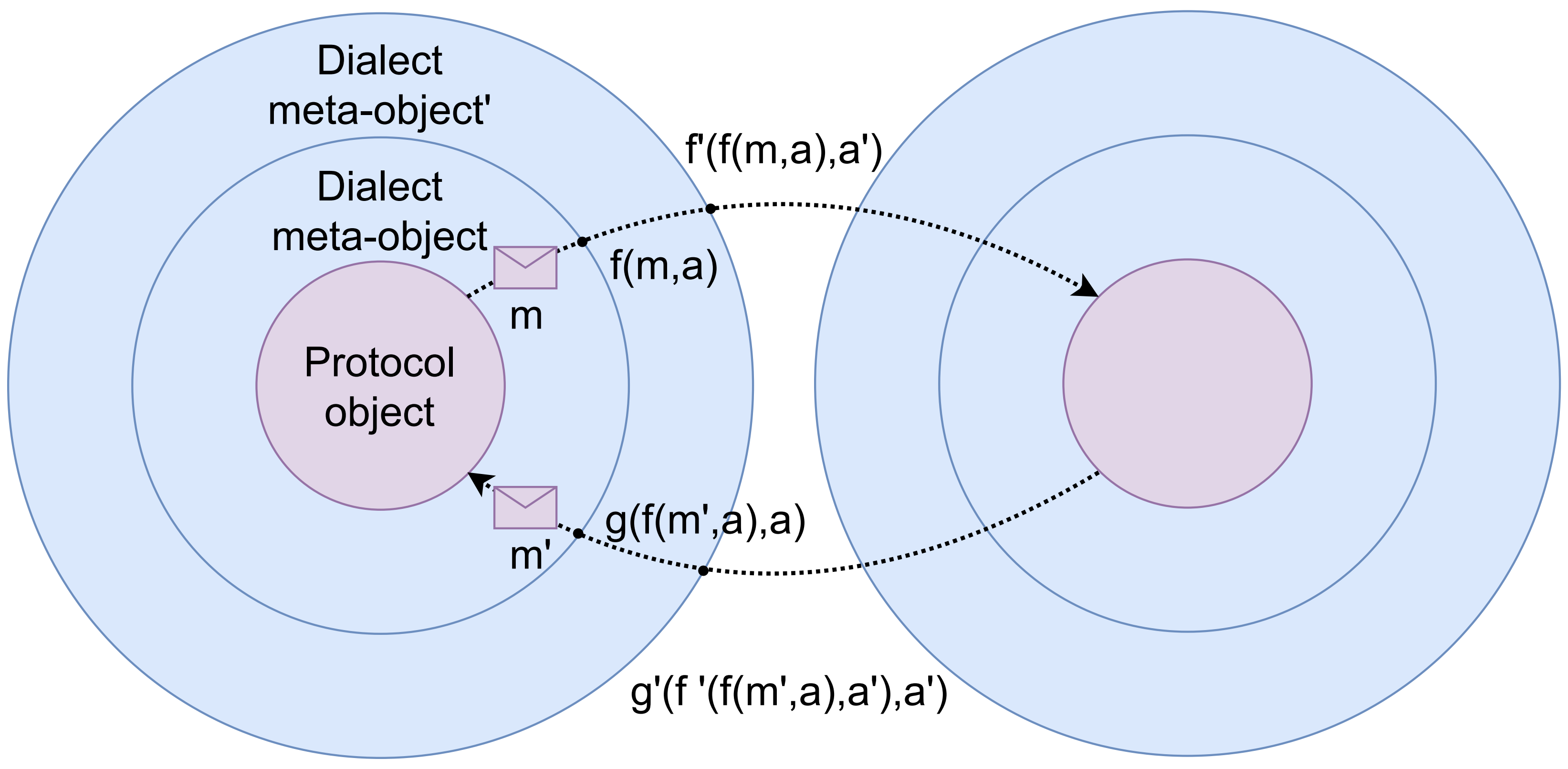
$Sh_i(m_i) \equiv Sh_{oi,pi}(m_i)$

Dialect Composition Operations

Our framework provides *dialect composition operations* to generate new, more sophisticated dialects out of simpler ones.

Vertical Composition

Vertical composition combines the transformations in a functional composition manner. From a meta-object point of view, vertical composition results in wrapping a second dialect meta-object on top of a first dialect meta-object. Of course, we can vertically compose not just two, but $n \geq 2$ dialects.



Horizontal Composition

Horizontal composition takes the lingos chosen by two dialects, merges them together, and uses a pseudo-random function to choose (with an optional bias) a lingo. As with vertical composition, this operation can be iterated to horizontally compose $n \geq 2$ dialects.

Conclusion

We have proposed modeling dialects as generic formal patterns, which transform protocols without changing their code. This is a vast generalization of the various dialects appeared so far in the literature that provides new methods to make dialects harder to break, e.g., by using dynamic dialects, and dialect composition operations.

Financiado por la Unión Europea
NextGenerationEU

Víctor García and Santiago Escobar were supported by INCIBE's Chair funded by the EU-NextGenerationEU through the Spanish government's Plan de Recuperación, Transformación y Resiliencia.