

# Timed Calculus of Cryptographic Communication

Johannes Borgström<sup>1</sup>, Olga Grinchtein<sup>2</sup>, and Simon Kramer<sup>3</sup>

<sup>1</sup> EECS, Technical University of Berlin

jobo@cs.tu-berlin.de

<sup>2</sup> IT, Uppsala University

olgag@it.uu.se

<sup>3</sup> Ecole Polytechnique Fédérale de Lausanne (EPFL)

simon.kramer@a3.epfl.ch

**Abstract.** We extend the (core) Calculus of Cryptographic Communication ( $C^3$ ) with real time, e.g., time stamps and timed keys. We illustrate how to use this extended calculus ( $tC^3$ ) on a specification and verification case study, namely the failure of the Wide-Mouthed-Frog protocol in its original, i.e., timed, version.

**Keywords:** Applied process calculi, timed cryptographic protocols, formal modelling, model-based specification and verification.

## 1 Introduction

Timed — as opposed to untimed — cryptographic protocols have received comparatively little attention from the formal methods community so far. The only timed formalisms for the modelling, specification, and verification of such protocols we are aware of are Timed CSP [1], tock-CSP [2], tCryptoSPA [3], the timed Spi-Calculus [4], and the (unnamed) process model from [5] (which we will refer to as tBEL). (Although Timed CSP and tock-CSP are special-purpose w.r.t. the temporal aspect, they — like core CSP — are actually not special-purpose w.r.t. the cryptographic aspect.) For practical usability, special-purpose models of timed cryptographic protocols are preferable over their general-purpose counterparts because (untimed) general-purpose models tend to create considerable (en)coding overhead: “[...] the coding up required would make the complex behaviour difficult to understand, and it is preferable to use a language designed to express such real-time behaviour.” [2].

In tock-CSP, tCryptoSPA, and timed Spi-Calculus time is *natural*-number valued. tock-CSP and tCryptoSPA provide local processes that globally synchronise through so-called tock events resp. tick actions, which represent the passage of one unit of time. And the timed Spi-Calculus provides a process constructor for querying a global clock. Thus, tock-CSP, tCryptoSPA, and the timed Spi-Calculus lack local clocks that potentially advance at different rates across different processes/locations. However [6], “[c]locks can become unsynchronized due to sabotage on or faults in the clocks or the synchronization mechanism, such as overflows and the dependence on potentially unreliable clocks on remote sites [...]”. Moreover [6], “[e]rroneous behaviors are generally expected during clock failures [...]” Hence, a faithful model of timed cryptographic protocols must allow for potentially desynchronised, local clocks.

In tBEL, time — in particular, a time stamp — is *real*-number valued, yielding a *dense* time domain. We contend that real-valued time-stamps are too fine-grained because protocol messages have finite length, which implies that real numbers are not transmittable as such. Moreover, real clocks only have finite precision. tBEL does provide local clocks, yet they “advance at the same rate as time.” [5, Page 2]. Further, adversarial break of short-term keys is modelled only indirectly with a parallel process rather than directly as part of the adversary model. Furthermore, tBEL lacks a process equivalence. On the other hand, tBEL comes with a (third-order) special-purpose logic for reasoning about tBEL models, and a decision procedure for a class of reachability properties of bounded protocols based on syntactic control points. In our opinion, tBEL and its associated logic are unnecessarily domain-specific. They seem to have been built from scratch rather than as Occham’s-razor extensions of untimed formalisms. Adding real-time to a model or logic without explicit time can be simple [7].

In contrast to the models discussed,  $tC^3$  extends  $C^3$  [8] with (1) *rational*-number valued time (which still is dense), (2) local clocks that may progress at different rates across different locations, and (3) adversarial break of short-term keys based on ciphertext-only attacks enabled by key expiration. Moreover,  $tC^3$  comes with a notion of observational process equivalence for model-based protocol specification and verification. As a property-based complement, we have also *co-designed* a logic, namely tCPL [9,10], for  $tC^3$ . The three primary features of the co-design of  $tC^3$  and tCPL are that (1)  $tC^3$ ’s notion of execution is a temporal accessibility relation for tCPL’s temporal modalities, (2)  $tC^3$ ’s notion of observational equivalence and tCPL’s notion of propositional knowledge have a common definitional basis, namely an epistemic accessibility relation defined in terms of structurally indistinguishable protocol histories, and (3) execution constraints of  $tC^3$ -processes are checkable via tCPL-satisfaction. These three features, especially Feature 2, are the result of our wholistic conception of model-based (process algebra) and property-based (modal logic) specification and verification as two truly complementary approaches. Other important features of  $C^3$ , and thus of  $tC^3$ , are explicit out-of-band communication and history-based key (and for  $tC^3$ , clock value) lookup.  $C^3$  neatly extends to  $tC^3$  by maintaining backwards compatibility. Essentially, only two additional axioms (and no modified axioms/rules!) are needed in its operational semantics.

## 2 Definition

Our timed Calculus of Cryptographic Communication is a conservative extension of a core calculus [8]. Core  $C^3$  consists of a language of distributed processes and an associated notion of concurrent execution in the style of structural operational semantics (SOS). Its *modelling hypotheses* are those of abstract ideal cryptography and interleaving concurrency. Cryptography is abstract in the sense that communicated information atoms (names) are logical constants and communicated information compounds are syntactic terms. We use pattern matching as a linguistic abstraction for cryptographic computation. Cryptography is ideal in the sense that cryptographic primitives are assumed to be perfect, i.e., unbreakable. Process execution engenders the activity of protocol participants and the standard Dolev-Yao adversary Eve, i.e., the generation of a