A Language and a Notion of Truth for Cryptographic Properties

Simon Kramer

École Polytechnique Fédérale de Lausanne, Switzerland
E-mail: Simon.Kramer@a3.epfl.ch

Motivation Protocol designers commonly specify a cryptographic protocol jointly by (1) a semi-formal *de*scription of its *behaviour* (local properties) in terms of *protocol narrations*, and by (2) an informal *pre*scription of its intended *goals* (global properties) in *natural language*. Informal specifications present three major drawbacks: (1) they do not have a well-defined, and thus a well-understood *meaning*; (2) they do not allow for the verification of *internal correctness* (referring to an internal notion of truth), i.e., the virtue that the conjunction of local properties implies each global property, typically by means of a *proof system*; and (3) they do not allow for the verification of *external correctness* (referring to an external notion of truth requiring a formal protocol model), i.e., the virtue that a proposed implementation (protocol model) satisfies each global property, typically by means of *model checking*.

In *formal* specifications of cryptographic protocols, local and global properties are expressed either explicitly *as such* in terms of a logical (or property-based) language, or implicitly *as code*, resp. *as encodings* in a protocol modelling (or model-based) language. Examples of such encodings are equations between instantiations of protocol schemata, and predicates defined inductively on the traces those instantiations may exhibit [1]. However, such encodings present four major drawbacks: (1) they have to be found; worse, (2) they may not even exist; (3) they are neither directly comparable with other encodings in the same or other protocol modelling languages, nor with properties expressed explicitly in terms of logical languages; and (4) they are difficult to understand because the intuition of the encoded property is implicit in the encoding.

Informal language and protocol modelling languages are patently inadequate for expressing and comparing cryptographic properties. It is our belief that only a logical language equipped with an appropriate notion of truth, i.e., a cryptographic *logic*, will produce the necessary adequacy therefore. A number of logics have been proposed in this aim so far, ranging from ad-hoc special-purpose cryptographic logics [2, the so-called BAN-logic] and [8, a unification of several BAN-logics], over varieties of classical modal and first-order logic used for the special purpose of cryptographic protocol analysis [4, temporal modalities], [5, epistemic modalities], and [6, deontic modalities], resp. [7, first-order], to combinations thereof, e.g., [3, epistemic post-conditions]. However in our opinion and w.r.t. our understanding of adequacy, each of these logics fails to be adequate due to limitations of *scope* (and *style*), i.e., the power to express (*intuitively*, *succinctly*, and *endogenously*[1]) arbitrary cryptographic goals, and / or *grain*, i.e., the power to discriminate sufficient detail in the analysis of cryptographic protocols. These limitations originate in *design decisions* of syntactical (language-defining *operators*) and / or semantic (meaning-defining *notion of truth*) nature.

Goal Our goal is to supply a logic that allows one to (1) *express* and *compare* arbitrary cryptographic properties intuitively, succinctly, and in an endogenous fashion, and to (2) *verify* correctness of cryptographic goals on cryptographic protocols up to a fine (though still *formalistic*) grain of detail. Our design decision thereby is to equip the logic with (1) four novel special-purpose basic operators and a selection of classical modal operators from *temporal*, *epistemic*, and *deontic* logics; and, in a first step, with (2) a novel, special-purpose external notion of truth defined through satisfaction in terms of models of *cryptographic processes*.

JUSTIFICATION A cryptographic protocol involves the concurrent interaction of participants that are physically separated by — and exchange messages across — an unreliable and insecure transmission medium. It is folklore that expressing properties of concurrent interaction requires temporal modalities. The physical separation by an unreliable and insecure transmission medium in turn demands the epistemic and deontic modalities. To see why, consider that the existence of such a separation and medium introduces an *uncertainty* among protocol participants about the *trustworthiness* of the execution of *communication acts* (sending and receiving) and the contents of exchanged *messages*,

[1] an endogenous (as opposed to exogenous) logical language is a purely property-based language. It is pure in the sense that the language is free from model-based forms, e.g., program fragments. Classical examples of endogenous and exogenous logical languages are LTL, CTL, and CTL*, resp. Hoare Logic and Dynamic Logic. The terms are due to Harel, Kozen, and Tyurin.

both w.r.t. *actuality* (an epistemic concern) and *legitimacy* (a deontic concern). (Note that it is exactly the role of a cryptographic protocol to re-establish this trustworthiness through the judicious use of *cryptographic evidence*, such as keys, hash values, and nonces.) We give priority to the definition of an external notion of truth because we opine that such a notion is practically more relevant, especially when defined through satisfaction in terms of a model of practically executable processes.

1. Language

Individuals (quantifiable) participant names (p, q, and r) and structured cryptographic messages (M and M').

Atomic (state) **predicates** the formulae $p \,\mathsf{k}\, M$, $\mathsf{s}(p, M, q)$, $p \,\mathsf{r}\, M$, and $M = M'$, pronounced 'p knows M', 'p sent M off to q', 'p received M', resp. 'M is syntact. equal to M''.

Compound predicates (1 state predicates) all atomic predicates and the formulae (ϕ and φ denoting state predicates) $\neg\phi$, $\phi \wedge \varphi$, $\forall v(\phi)$, $\mathsf{K}_p(\phi)$, $\mathsf{P}(\phi)$, and $\forall\phi$, pronounced 'not ϕ', 'ϕ and φ', 'for all v, ϕ', 'p knows that ϕ', 'it is permitted that ϕ', resp. 'for all futures, ϕ'; and (2 path predicates) all comp. state predicates and the formulae (ϕ and φ denoting path predicates): $\neg\phi$, $\phi \wedge \varphi$, $\forall v(\phi)$, $\mathsf{K}_p(\phi)$, $\mathsf{P}(\phi)$, $\phi \,\mathsf{B}\, \varphi$, $\ominus\phi$, $\oplus\phi$, and $\phi \,\mathsf{W}\, \varphi$, the latter four being pronounced 'ϕ back to φ', 'previously ϕ', 'next ϕ', resp. 'ϕ waiting for φ'.

DISCUSSION The temporal fragment of our language coincides with the syntax of a form of CTL*. Further, we introduce a novel operator (k) for first-order (or knowledge *de re*, i.e., knowledge *of* objects) and adopt the operators K_p for higher-order knowledge (or knowledge *de dicto*, i.e., knowledge of facts *about* objects) from classical epistemic logics. First-order knowledge shall convey *possession* and *understanding of the purpose* of a piece of cryptographic information *up to* cryptographically irreducible parts. We adopt the operator P from classical deontic logics as basic because permission coincides with *authorisation*, which is essential for modelling legitimacy.

2. Notion of truth

Model It is a generic (arbitrary message language), nominal (communication through matching participant names) calculus of communicating guarded (guard language = logical language) processes. We define an example message language $\mathcal{M}$ with primitives for tupling, symmetric and asymmetric encryption, signature creation, and hashing. Participant-name-based communication makes the translation of protocol narrations to process models almost a task of transcription. Coinciding guard and logical languages have the advantage that protocols become proof-carrying code amenable to run-time verification. We annotate processes P to form triples $\mathfrak{s} := \mathcal{K}_{\mathtt{A}} \circ P : [\mathcal{K}^-, \mathcal{K}^+] \circ \mathcal{H}$, where $\mathcal{K}_{\mathtt{A}}$, $\mathcal{K}^-$, and $\mathcal{K}^+$ denote sets of first-order adversary, resp. private and public participant knowledge, and $\mathcal{H}$ denotes a history of communication acts. The reduction relation models protocol execution, i.e., the activity of protocol participants and the (Dolev-Yao) *adversary* ($\mathtt{A}$), and the evolution (involving *computation*) of their respective knowledge.

Satisfaction We define it by *nested* induction on the structure of formulae around satisfaction of CTL*.

3. Cryptographic goals

$\mathfrak{p}@i \vDash_{\mathfrak{M}} \forall \boxplus \neg \mathtt{A} \,\mathsf{k}\, M'$	secrecy
$\mathfrak{p}@i \vDash_{\mathfrak{M}} \mathsf{K}_q(\mathsf{s}(p, M, q))$	authenticity
$\mathfrak{p}@i \vDash_{\mathfrak{M}} \mathsf{s}(p, M, q) \rightarrow \mathsf{K}_q(\mathsf{s}(p, M, q))$	non-repudiation
$\mathfrak{p}@i \vDash_{\mathfrak{M}} \mathsf{s}(p, M, q) \rightarrow \mathsf{K}_p(q \,\mathsf{r}\, M)$	non-repudiation
$\mathfrak{p}@i \vDash_{\mathfrak{M}} \mathsf{K}_q(\mathsf{P}(\psi(r)))$	authorisation
$\mathfrak{p}@i \vDash_{\mathfrak{M}} \exists v(\psi \rightarrow \forall \boxplus \neg \mathsf{K}_q(\psi))$	anonymity

where $\mathfrak{p}$ denotes a path with state $\mathfrak{s}$ in position i; $\mathfrak{M} := (\mathcal{M}, \Omega)$ with Ω denoting a set of operations for detupling, decryption, and signature verification; $M' \in \mathcal{M}$ denotes a confidential message, i.e., $\mathfrak{p}@i \vDash_{\mathfrak{M}} \mathsf{F}(\mathtt{A} \,\mathsf{k}\, M')$, where $\mathsf{F}(\phi) := \neg\mathsf{P}(\phi)$ is pronounced 'it is forbidden that ϕ'; and $\boxplus\phi := \phi \,\mathsf{W}\, \bot$ and $\psi(v) ::= \mathsf{s}(p, M, v) \mid p \,\mathsf{r}\, M$. We suggest comparing cryptographic goals by a relation of *semantic consequence* ($\Rightarrow$), defined s.t. $\phi \Rightarrow \varphi$:iff for all $\mathfrak{p}$ and i, if $\mathfrak{p}@i \vDash_{\mathfrak{M}} \phi$ then $\mathfrak{p}@i \vDash_{\mathfrak{M}} \varphi$ (inducing a *lattice* of cryptographic goals).

References

[1] M. Abadi. Security protocols and their properties. In *Foundations of Secure Computation*, volume 175 of *NATO Science Series: Computer & Systems Sciences*. IOS Press, 2000.

[2] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1), 1990.

[3] N. Durgin, J. Mitchell, and D. Pavlovic. A compositional logic for proving security properties of protocols. 2002.

[4] J. W. Gray and J. D. McLean. Using temporal logic to specify and verify cryptographic protocols (progress report). In *8th IEEE Computer Security Foundations Workshop*. IEEE Computer Society Press, 1995.

[5] J. Halpern and K. O'Neill. Secrecy in multi-agent systems. IEEE Computer Security Foundations Workshop. IEEE Computer Society Press, 2002.

[6] J.-J. C. Meyer and R. J. Wieringa, editors. *Deontic Logic in Computer Science: Normative System Specification*. John Wiley & Sons, 1993.

[7] L. C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6(1), 1998.

[8] P. F. Syverson and P. C. van Oorschot. A unified cryptographic protocol logic. NRL CHAC 5540-227, Naval Research Laboratory, Center for High Assurance Computer Systems, Washington D.C., USA, 1996.