Dear Editor,

Please find in the following my comments on authors' revision.

Importantly, the aspects regarding the clarification of the contribute of the paper have been considered and resolved with the editors.
Being that my many concern, I have a few further comments on authors' reply, but these do not need a further round of revision.
However, I believe that addressing them could still improve the paper and the required example and information should be provided before publication.

First and foremost, we should like to thank the anonymous referees for their effort in reviewing of our work and for their useful suggestions. In the following, we go through the referees\' comments and show how they have been taken into account in revising the paper. Unless otherwise stated, we have also corrected all the typos as suggested by the referees.

Summary

A framework for the static verification of security properties is presented in this paper. Systems of interest are expressed in the Quality Calculus QC, inspired by the Pi-calculus, and security properties are expressed in terms of knowledge/accessibility to (unstructured) names/channels, in a reachability-like formulation.

The QC specification of a system can be translated into logical formulas that include the security property to be validated, i.e. a label-based reachability statement. An attack is a model of the formulas, that is, an assignment that highlights the information needed to carry out the attack. The attacker model consists of an interacting participant capable to guess any secret (channel) at a cost, and drive the system to the "execution" of a desired label. The proposed static analysis returns an under-approximation of the possible attacks.

A qualitative/quantitative cost structure is associated to the system specification. Reachable labels have associated a security level (e.g., high, low, … - a lattice, as standard), which can be somehow determined from quantitative measures of label's protection. Analogously, attacks have have a cost - the cost of guessing secrets - that can be also mapped to a security level. System's safety is understood as the non-existence — up to the limits of the under-approximation! — of an attack that is "cheaper" than the security level expected for a given label (otherwise, informally speaking, "one can, too easily !, break a presumed safe feature of the system").

Attack/model discovery is performed by optimisation/satisfaction modulo theories which finds attacks of minimal (in the lattice) cost.

From the logical interpretation of the system, it is possible to automatically derive and represent attacks as 'attack trees', a visual/graphical formalism.

A simple running example illustrates the theory. A prototype tool is referenced.

The paper is interesting and within the scope of the journal, but there are some points that should be clarified (reported below).

However, my main issue with this submission, an extended version of [42], is a substantial overlap with already published material by the same authors, particularly [41]. The proposed analysis — QC and system description, static analysis, quantitative attack evaluation, running example, prototype tools, and attack tree representation — seems to be closely related to what presented in [41]. There are some (relevant?) differences, e.g. the explicit formalisation of guessing secrets, and the definition of the expected level of safety for system's label, but, nonetheless, for instance, the same minimal attack and its precise cost is detected in the same running example in both papers, hence same quantitative results. It seems that the original contribution of the paper should be further clarified/expanded before publication.

I opted for "The paper can possibly be accepted for Logical Methods in Computer Science. Another refereeing round is needed. " amongst the suggested recommendations, but I would leave with the editors the decision on whether either the paper contribution is/is not sufficiently original for the journal standards, or it could become so once that the paper will be suitably expanded (possibly also addressing the open issues below).

Further comments (and typos) —

- Highlighting the specific contributions of the paper — especially wrt [41] — could facilitate the reading/understanding of the paper,

We have expanded the relevant paragraph of the introduction.

- The class of possible attacks that the framework is able to detect should be clarified - possibly by means of examples. How expressive are flat channels? Possibly, an attack like Lowe's one, i.e. where the structure of messages is relevant, could not be detected. Would it be possible to provide further examples of detectable attacks/systems of interest (particularly, cyber-physical systems are repeatedly mentioned, why is the framework particularly suitable for them? Could a convincing example be provided)?

We agree with the referee that there is a strict relationship between classic security protocol verification techniques and the framework we presented, as we acknowledged in our survey of related work. We sketch in Appendix C how the framework can be extended to modelling and analysing structured channels (messages). Nonetheless, as we observed in the introduction and in §2.1, we

deliberately chose to shape the paper around a propositional analysis and thus resort to flat channels, for we believe they serve different modelling purposes and different analysis needs. Security protocols looks like tiny artefacts when contrasted with distributed systems, the latter using typically many protocols as sub-routines. In this light, security protocols can be understood as basic building blocks that we would like to prove flawless; to this end, modelling the structure of messages and the functions operating over them is necessary (and often not enough). On the other hand, such a low-level representation seems instead not suitable to modelling complex systems and to support analysis whose results that can be computable in practice and presented in a human-readable format. In this sense, the coarse abstraction we adopt with flat channels is the price to pay for scaling up from analysing protocols to analysing complex systems, where a flat name can represent an entire sub-system (software, physical, cyber-physical) whose analysis is condensed in the price/security guarantee attached to it.
We have extended the last paragraph of §2.1 promoting the above considerations.

> - Analogous considerations hold for the under approximation induced by the static analysis and the attacker's model. Could they be better characterised? E.g. which other measure of the difficulty of guessing secrets could be appropriate? What does it mean that no attack can be detected for a given label (even when the attacker can guess any secret !)? What kinds and other examples of attacks can be modelled? …

We believe that the under-approximation induced by the analysis is described to a proper level of formality in the text (cf. §4.3) and substantiated in Appendix A. In the attack discovery procedure (qualitative analysis) there is no attack when the label of interest is unreachable (cf. § 4.3). In the attack quantification procedure, no attack to a given label means no weak path to that label, that is, in order to reach the label an attacker must pay a price corresponding to the security level associated with the label. In other words, locking your car is useless if you leave the window open: there exists a path leading to entering the car without bothering with the lock. Similarly, you cannot complain if the NSA breaks your 512-bit RSA key.
However, we understand the referee uneasiness with the vagueness connected to the modelling power of flat channels, and we think the paragraph responding to the previous point also addresses this concern.

I am fine with the trade-off between expressiveness and efficiency that can be represented by the choice of focussing on flat channels. However, I still believe that the claim that this is a suitable choice to model a wide range of different system should be further justified, particularly when cyber-physical systems are concerned, as their properties, for instance, encompass and rely upon physical values - continuous variables, say - whose treatment usually pose a whole new set of different challenges. I am not saying that the framework cannot cope with cyber-physical systems, but I'd like to see a significant example supporting the claim.

I suggest to make the links to cyber-physical system more clear. An example of the analysis of an attack to a relevant property of a cyber-physical systems should be provided in the paper.

> - The definition of the expected level of safety for system's label seems to be a bit problematic and arbitrary. Doesn't it depend, in the example,

> on the knowledge of the efforts required to attack it (p.23)? Is this information always available before detecting attacks? But such choice can affect the results of the analysis itself (and the found attacks), can't it?. Are there other ways of defining the expected level of safety? How should it be done?

We agree with the referee that it is not always possible to get a quantitative characterisation of security. Nonetheless, we believe that the elegance of our framework partly lies in the modularity of the attack discovery and quantification procedures: whenever a quantitative characterisation is not available, we can still provide the set of all potential attacks and represent it graphically! The definition of cost structures for security problems is a research areas in itself, and we rely on the cost map as an input to the analysis. We have shown how information theoretic arguments can be exploited to come up with numerical cost maps. At the same time, as the referee has noticed in the example, the analysis requires to define a connection between the cost structure for channels and the security levels for labels. To facilitate this intrinsically difficult task, the framework allows using partially-ordered, symbolic maps (cf. §5.5).

> - Further experimental results (efficiency?), could facilitate the understanding of the framework and its relevance and viability (please state the differences amongst the two tools in [41] and in this paper)

We have added a new sentence in the tool section (§ 7.2) to clarify the scope of the tool presented in this paper with respect to the tool of [41]. As for the efficiency issue, we have already observed in §7.1 that the study of performance in solving satisfiability problem is a research area on its own, and we do not want to risk enticing the reader in generalization we cannot support.

Given the complexity of the several components involved, the suggestion was indeed about experimental, but still informative, results.

How much does it take to detect the attack in the example?
How much does it scale up? What if you have 10 - 100 - 1000 users? (or any other way to scale it up you may prefer).
What is the largest system (or instance of the mentioned example) that you have been able to validate?
...

Some such information should be added to the paper (to the section about the tool, perhaps).

> p.4 +5 allows US
> p.5 to compute_ing
> …