Security Protocols as Environments: a Lesson from Non-collaboration

Maria-Camilla Fiazza Michele Peroli Luca Viganò
Department of Computer Science
University of Verona, Italy
Email: tokhami@gmail.com, michele.peroli@univr.it, luca.vigano@univr.it

***Abstract*—Although computer security typically revolves around threats, attacks and defenses, the sub-field of security protocol analysis (SPA) has so far focused almost exclusively on attacks. In this paper, we show that such focus on attacks depends on few critical assumptions that have been characteristic of the field and have governed its mindset, approach and developed tools. We motivate that indeed there is room in SPA for a fruitful notion of defense and that the conceptual bridge lies in multiple non-collaborating attackers. Defending security protocols through interference between attackers is possible; however, in order to understand network behavior completely, it is necessary to start treating protocols as environments, not simply as sequences of message exchanges.**

***Index Terms*—Collaboration and non-collaboration, Defense, Environment, Multiple non-collaborating attackers, Security guardian, Security protocol analysis.**

I. Introduction

A large number of logics, formalisms, and tools have been proposed in recent years for formalizing and reasoning about security protocols or services; see, e.g., [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14] just to name a few and see [15], [16] for a simple and detailed introduction to security protocols and *security protocol analysis (SPA)*. The typical attacker model adopted in SPA is the *Dolev-Yao (DY) attacker* [17], who can compose, send and intercept messages at will, but, following the perfect cryptography assumption, cannot break cryptography. The DY attacker is thus in complete control of the network—in fact, he is often formalized as being the network itself—and, with respect to network abilities, he is actually stronger than any attacker that can be implemented in real-life situations. Hence, if a protocol is proved to be secure under the DY attacker, it will also withstand attacks carried out by less powerful attackers; aside from deviations from the specification introduced in the implementation phase, the protocol can thus be safely employed in real-life networks, at least in principle.

It can be shown that analysis models with *multiple collaborating DY attackers* are not interesting, since they can be quite straightforwardly reduced to the simpler model with just one DY attacker (see, e.g., [18] for a detailed proof, as well as [19], [20], [21] for general results on the reduction of the number of agents to be considered). The situation is completely different, and much more interesting, when one considers *multiple non-collaborating DY attackers*: when independent Dolev-Yao attackers act simultaneously, they may fail to break a vulnerable security protocol even though a single attacker will always succeed. In terms of attack power, a system composed of non-collaborating DY attackers is *weaker* than a single DY attacker. Attackers in this scenario can fail because of *attack interference*, a phenomenon that cannot be explored in the classical setting of SPA—a field which has specialized its approach and tools to address the *existence* of attacks against a given protocol.

The emergence of interference between simultaneous attack procedures brings to the forefront the open question of which factors concur to determine the success of an attacker at the expense of another. Clearly, studying interference requires more general network and agent models than those in classical settings; the network model must at least be able to handle concurrency and the agent model must support at least contextual decisional procedures.

We provided one such model in our exploratory work [22] and its extended version [23], where we described two case studies that illustrate how it is possible to exploit attack interference to mitigate protocol vulnerabilities, thus providing a form of protection to security protocols. In this paper, we go one big step further (or, in a sense, take a step back) towards a more general and mature view on what non-collaboration has to offer regarding SPA. Although our non-collaborative model was originally constructed as a generalization of the classical SPA set-up, we enucleate here how its implications open up an entirely new perspective on how to think about security protocols. At the heart of this new perspective lies the idea that security protocols can be best understood in terms of *environments*, rather than simply as sequences of instructions to norm message exchanges. This novel idea and its implications are the main contributions of this paper.

The paper is organized as follows: in Section II, we cover the tight relationship between security protocols and sets of interactions, showing that interference gives rise to novel properties of interest, that attackers can build a representation of the protocol in terms of states with such properties and that interference can be exploited to construct defenses; we advocate that reactions should be included within the protocol model. In Section III, we cover the relationship between the notion of security and attacker capability in two classes of approaches (classical SPA and economic approaches), showing that trade-offs are intrinsic in security protocols whenever attack interference can occur. In both sections, we draw the

COLLABORATECOM 2011, October 15-18, Orlando, United States

DOI 10.4108/icst.collaboratecom.2011.247156

same conclusion: security protocols are best understood as environments. Finally, in Section IV, we summarize our novel paradigm and discuss future work.

II. From the classical approach to an environmental paradigm for SPA

In general (and in a nutshell), a security protocol P is a sequence of messages exchanged by a number of agents in order to obtain a security property π_P (or several such properties at the same time, e.g., confidentiality of data, authentication of messages and agents, etc.). The property π_P can be formalized as a logical formula that must hold true for all possible executions of P under the presence of an adversary E, which can in turn be formalized in terms of an inference system. Two factors complete the picture: (i) a suitable logical formalism, such as (multi-)set rewriting [1], [14] or distributed temporal logic [19], just to name a few of the several possibilities, and (ii) a strategy to reduce and handle the staggering complexity of the resultant search tree. If a deduction path is found in which π_P fails to hold, then P is vulnerable to the attack corresponding to the path, which we will indicate in the following as A_P.

In our exploratory work, we have examined, from the classical SPA viewpoint, the case of multiple non-collaborating attackers, simultaneously mounting attacks against the same protocol run. In this setting, it is natural to generalize the outcome of an attack procedure, moving from binary success/failure to a more complex panorama. Attacker E may indeed acquire a message m that is meant to be protected under P, as it violates π_P; however, the attacker may not recognize m as such, having received from competitors misleading messages of the same type, which pollute E's knowledge and decisions and cause the identification of m to involve guesswork.[1]

The recognizability of the "success message" m is a property of the *message*; it depends on the specific network behaviors of all attackers and is not derivable solely from the attack A_P. The attack A_P against P is the classical attack procedure discovered by SPA techniques. We consider it a *base attack*, because it captures only the interactions between the attacker and the honest agents under attack. In non-collaborative settings, the base attack is only half of the story: there are multiple ways to interact with other dishonest agents while implementing the same base attack against honest agents. We have explored "extended" attack procedures that correspond to competitive attackers, i.e., attackers who attempt to gain exclusive success in violating the security property π_P.

In non-collaborative settings, in addition to properties of the "success" messages, there are also properties that can be traced directly to the pair (protocol P, attack A_P) and that characterize successful types of behaviors; one such property of interest is *exclusivity of success*. If A_P is based on the attacker replacing some information with his own, then it is apparent that no two attackers can succeed in the same protocol run; in order to succeed, attackers have to gain an advantage over their competitors and ensure a degree of dominance (see, e.g., the analysis of the BME protocol in [22]).

[1]See the analysis of the SRA3P protocol in [23] for more details. Note also that an attack to a security protocol may, of course, be characterized by a more complex situation than just the confidentiality of a single message m; for instance, in the case of mutual authentication, several messages are considered. Our considerations in this paper are general and hold also for more complex attack situations, although we treat just the case of message m for brevity and simplicity.

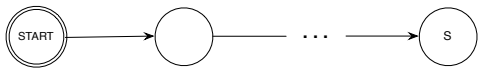

Fig. 1. Worldview of P, as seen by a classical attacker in isolation.

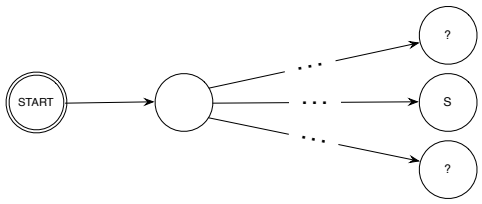

Fig. 2. Worldview of P, as seen by a classical attacker immersed in an interactive environment with other classical attackers.

These simple examples illustrate that there are a number of *properties* of the message exchange that relate to the attacker's success—and are thus relevant for security purposes—but that are not customarily taken into consideration in the traditional one-attacker set-up of SPA. From the classical point of view, the only relevant factor is the existence (derivability in at least one circumstance) of a message or a set of messages that violate the security property meant to be upheld through P.

A. The attacker's worldview

The significance of additional security properties is apparent to the researcher, but it is also visible to attackers. Let us take the perspective of an attacker and build the scenario progressively. Initially, attacker E is in the classical setting: alone against a set of honest agents. The worldview of the attacker is shown in Fig. 1; the relevant steps lend themselves to being interpreted very simply as: "When I witness the opening message of P, I start the attack procedure A_P and I reach the success state S".

As soon as the classical attacker E is placed in a non-collaborative setting with a classical competitor, attack interference causes the attack procedure A_P to result in a range of outcomes, only one of which is the success state S (Fig. 2). The attacker can realize that the end state is not the expected one when he attempts to use (post-protocol) the information he has acquired as a result of the attack; at times, it may happen that attackers can realize their failure directly during the protocol run, as they may fail to acquire *any* message that can violate π_P.

Attackers can then examine these unexpected states and realize that the significant feature they describe is either partial success or downright failure. In this step, we (and attackers

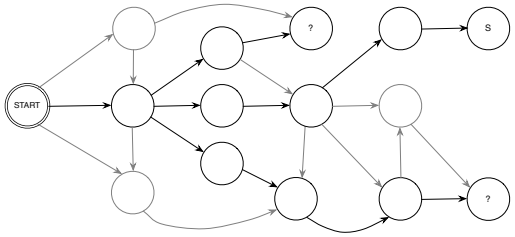

Fig. 3. Worldview of P, as seen by a competitive attacker immersed in an interactive environment with other competitive attackers.

along with us) are performing a great deal of abstraction, translating a low-level description (state of the attacker's knowledge base) to high-level knowledge (partial success, failure). This step is in fact not different from what is done in classical SPA: inference on the non-derivability of the messages violating π_P in states different from S.

The attacker, now aware that the worldview in Fig. 2 does not adhere to his expectations, will select the states that best correspond to his goals and seek to devise a new attack behavior, better suited to the actual circumstances. The attacker is now a *competitive* attacker in a non-collaborative world.

Competitive attackers devise variants to their attack behavior, which could result in success, complete or at least partial. Depending on the features of the protocol P and of the base attack, outcomes differ. They can be ranked from the point of view of honest agents and rank may be taken as a measure of attack severity. Little by little, our attackers build a map between the attack (variants) they use, the conditions in which they operate and the actual security "result" they can reach. By matching the worldview to states reached during sequences of non-collaborative runs of the protocol, attackers can recognize unexpected states and uncover further nodes, thus progressively uncovering the structure of the interactions that can occur under P (Fig. 3).

The worldview will keep expanding as different types of interactions emerge, as consequence of a range of different strategies, attack variants and base attacks (in case P admits more than one). However, we do not expect the worldview to expand indefinitely: its complexity is bound by P's. The actual size of the worldview depends on the amount of abstraction that we have been able to perform: if we had kept the analysis at the message level the structural complexity of the worldview, even for toy protocols (such as those in [16]), would be immense.

B. Security protocols as environments

The expanding worldview of non-collaborative attackers is, in fact, their understanding of P, i.e., the understanding of what can occur under P. Rather than as a behavioral specification for honest agents, P becomes of interest in terms of the set of interactions it admits and their properties.

The classical view of P is as something that describes and norms a sequence of actions (message exchanges); the type of abstraction it admits is limited to success or failure in deriving a message that violates π_P. Non-collaborative settings highlight that it would be highly beneficial to evolve our understanding of P, viewing it as something that both *describes and norms the possible interactions* that can occur under it. By capturing sets of interactions, P implicitly also describes and norms the set of possible security results and the range of behaviors that lead to the same outcome. All this is effectively summarized in the P-worldview.

Abstraction is much richer; it starts from identifying properties of the worldview states that are relevant when reasoning about the security outcomes of the interaction. Once a property ϕ of interest is identified, then analysis of P (seen as a worldview) expands into an innovative set of questions. These questions prompt a parallel abstraction—this time on base attacks and on protocols-as-message-exchanges:

1) Which features of the protocol P and which features of the base attack A_P contribute to the existence of final states in which ϕ holds?
2) Which features of the competitive attack behavior are involved in enforcing ϕ on opponents or in gaining ϕ for oneself?

We have shown with the toy protocol BME in [22] that knowledge of the identity of a competitor is a relevant factor for the success of an attack. Success criteria for attackers should now explicitly include the possibility of detecting previously unknown competitors and should account for the possibility that messages violating the security property may be derivable but not recognizable with certainty. In short, in non-collaborative settings, success criteria must be as complex as the types of situations emerging from the interactions.

Once we immerse attacker E in an environment in which the presence of competitors can have a negative effect on E's success, it is only natural for E to devise strategies suited to protect his own attack from interference. Studying P for attack identification (as classical SPA is fully equipped to do) can yield the *base attack*, but a complex execution environment will lead to *variants* of the attack behavior. The search for variants is not supported by SPA machinery. The variants enforce the same interaction with respect to honest agents, but capture different interactions with respect to the additional attackers. SPA has not yet developed systematic search tools for multi-attacker situations.

The overall conceptual panorama is veering towards fields, such as AI, that have a complex notion of agent, as opposed to the weak notion of "agent playing in a role" that has been characteristic of SPA. Even though our perspective on security protocols is closer to exploring a structure than devising strategies for game-theoretic "battles", we believe that adapting mature tools from AI or robotics to analyze non-collaboration may be a fruitful direction of future research.

The main point so far is that P is actually describing, norming and regulating a set of admissible interactions, which correspond to the possible behaviors of honest and dishonest agents exchanging information in accordance to P. In addition, some message properties, which turn out to be relevant for the

security outcome of the protocol run, emerge as a result of the interaction, without having to be known explicitly to the agents nor having to be explicitly computed. Attackers find themselves immersed in a complex mechanism that demands systemic thinking and abstraction for successful adaptation.

We find that this is a very fitting description for the notion of *environment* and we propose that protocols be viewed in terms of environments characterized by a corresponding "law". P, rather than something to follow, appears to be something to *explore*.

C. Interference as defense: introducing a guardian agent

SPA has been developing an extensive body of knowledge on the mechanism by which the intended functioning of a protocol is disrupted. Our understanding of protocols, on the other hand, has not been growing at the same rate.

The phenomenon of interference between non-collaborating attackers shows that the execution environment in which an attack occurs matters—and it matters so much that it makes the difference between an attack succeeding always and succeeding only sometimes. Thus, defense against an attack can be realized by manipulating the execution environment. We have shown in [22] that a partial defense can be derived by placing in the network environment an additional agent (that we call *guardian agent*) that behaves as a competitive attacker and interferes with attacks originating from agents with actual malicious intentions.

In a sense, it is sufficient to introduce a benign network agent to turn the tables on the invincible attacker with a sure-fire attack against a vulnerable protocol. Naturally, the attack will sometimes succeed, because P *is* vulnerable and E knows an attack trace against it. This finding is interesting because it shows, under yet a different perspective, that attack identification for P is far from being the whole story for what concerns the intrinsic security of P.

The concept of interference can be pushed a little further to yield a second layer of (partial) defenses. In particular security settings (key agreement or key exchange protocols [15], [16]), violation of a protocol results in a security failure only if honest agents are unaware that the key is now known to an attacker. In these circumstances, detection of ongoing malicious activity protects the user just as much as prevention of ongoing malicious activity. The guardian agent can also serve as means to notify honest agents of ongoing attacks: if the honest agent is capable of detecting deviations from the normal execution of the protocol steps, then the anomalous activity caused by the presence of multiple active attackers can be used to construct defenses.

One may design the behavior of the guardian agent as an agent collaborating with the honest agents, with the specific intention of raising *warning flags* to which honest agents can pay attention and react. The guardian agent is, to all effects and purposes, a benign attacker, trusted by the honest agents. Honest agents should not, in principle, be informed of the specific attack trace to which they are vulnerable. Hence, if they can detect malicious activity at all, it has to be on the basis of flags that are independent of the specific attack trace—and, in general, independent also of the protocol in use. Such flags encode *local* defense criteria and can be as simple as realizing that no answer has arrived within a time considered reasonable or realizing that two (possibly different) answers have been sent in response to a single request.

Note that in our novel view we are not advocating a "battle of wits" between the attacker and the guardian—a battle with unknown results, as it depends also on the (re)actions of the honest agents; rather, we are describing a dynamic view of protocols, which emerges only when considering competing (and thus "battle") situations, with attacks, attack interference and defenses. Guardian-based defenses, realized through the exploitation of attack interference, find a practical application in the time lag between the discovery of a vulnerability in a protocol already in use and the distribution of a fix—either in the form of a patch or by switching to a safer protocol. Historically, SPA has been successful in finding attacks to a number of security protocols in use, including widely employed protocols such as Kerberos or Single-Sign On (see, e.g., [24], [25], [26]), and it would thus be extremely beneficial to have defense means that allow users to provisionally tolerate and keep on using vulnerable deployed protocols while the protocol designers and engineers work at deploying corrected versions.

D. Including reactions within the scope of the protocol model

Honest agents that pay attention to simple flags of ongoing attacks and attackers that pay attention to the degree to which they fail their goals require moving away from agents as simple routines and towards richer notions of agents. To do so, the machinery must be adapted to support a number of features that are currently either unavailable or incompatible with the tools developed under the one-attacker assumption.

Standard models for agents, formalized in terms of inference systems, rely on the protocol's prescriptions to such a degree that even network primitives are affected. In fact, classical (honest) agents are often *unable* to receive messages that do not conform to expectations, as specified by the protocol. In the classical set-up, rules related to sending and receiving messages are applied only in the exact order in which they appear in the role specification (or in the attack trace); if so, agents cannot receive messages that they do not expect. It has been shown that unrestricted reapplication of these rules could lead to identifying "false attacks".

This hard-wired lack of flexibility makes classical agents inescapably stupid. There is no room for a strategy because there is no room for any other behavior than that described by a single sequence of states or steps. This limitation is in line with the traditional understanding of protocols as sequences of message exchanges. Under this perspective, classical agents are finite state machines with only one path of transitions. The model is very handy for governing complexity when searching for attacks, but it is very limiting when attempting to construct agents that can pay attention to simple anomalous network activity.

A cornerstone of our approach consists in extending the usual notions of protocol and role by introducing a control—a mechanism to regulate the execution of the steps prescribed by the attack trace P_A in accordance with the attacker's chosen competitive strategy. By introducing agent controllers, we support agents that can react to their context; we also open the way to selective and contextual rule reapplication. In [22], this step was implemented implicitly, by providing a detailed description of the agent's behavior and decisions.

In our model, honest agents perform a controlled execution of the protocol as well, so as to support in-protocol detection of attacks. Honest agents behave according to the protocol's prescription, expect things to go exactly in accordance with the protocol and interpret deviations in terms of the activity of dishonest agents. This "operational" definition of honest agents is rather different than the classical view, in which honest agents are those that carry out *only* the (explicit and implicit) actions specified by the protocols.

The disagreement between the traditional view of honest agents and the view we propose is rooted in different understandings of what a protocol is. The notation in use to specify protocols suffers from a flaw of incompleteness: when agents cannot receive unexpected messages, it is not possible to specify a policy to react to errors. Reaction to or recovery from error is considered as something that takes place after the protocol has failed and the corresponding run or session has been closed: rather than a part of the protocol, it is a part of the post-protocol world. We believe it should be considered, instead, as a part of the in-protocol world: reaction to error or other unexpected events belong to the range of behaviors and interactions normed by P. In doing so, honest agents are not doing any differently than attackers realizing, during the protocol run, that their attack is not going as expected and taking countermeasures against their competitors. One may argue that honest agents should, by default, be kept from having the same abilities as attackers; we wish to point out that what is involved here is the notion of *agent*, rather than its network abilities (spying, erasing or injecting messages).

We propose that specific reactions and defenses be included within the scope of the protocol model, thus preserving a coherent view of P as an environment for both honest and dishonest agents.

III. Security: invulnerability or compromise?

In this section, we present a novel orthogonal perspective on protocols, focusing on the connections between the notion of security and agent capabilities. We cover the perspective of classical SPA and, in more detail, that of recent economic approaches such as retaliation.[2] We conclude the section by discussing how our environmental take on non-collaborative scenarios relates to these classes of approaches: we show that interference induces trade-offs *within* P, thus strengthening the reasoning that protocols can fruitfully be viewed as environments.

A. Security as invulnerability

In the classical setting, as soon as an attack procedure A_P is known, a single attacker with knowledge of A_P will deterministically succeed in attacking P. At this point, the only reasonable interpretation of security is the absence of attacks that can be known to dishonest agents. In fact, in this context, security means invulnerability.

This reasoning also governs the emphasis on attacker capabilities as the key measure of security: if P is proved to be secure under an unrealistically strong attacker (the DY attacker), it will also withstand attacks carried out by less powerful attackers; the protocol can thus be safely employed in real-life networks. Again, security is equated with invulnerability. Invulnerability against attacks (via partial property-preservation) is tacitly assumed as a goal even in approaches that explicitly consider vulnerability (such as [27], [28]). In a sense, we are dealing with a sophisticated notion of invulnerability by (partial) attack-indifference: compromise has limited repercussions and the properties of interest of the protocol are those reliable even under partial compromise.

When an attack is discovered for protocol P, P loses its status as a "proper" protocol. If the discovery is at the hands of a security engineer, then P is dismissed as a candidate protocol and redesigned to address its weaknesses. If the vulnerability is discovered first by a malicious agent, then it can be exploited—until a security engineer becomes aware of it. In all cases, when the "good guys" discover that a given protocol P is vulnerable, the search for a better protocol starts. This story is the only possible story if we are investigating P independently of its deployment environment.

Is it even possible to design a "permanent defense" through invulnerability? Real life scenarios seem to give no as the clear answer. Discovery of vulnerabilities often occurs well after deployment, in some cases even years after deployment. At times, vulnerabilities that are not present in the protocol are introduced at the implementation phase; in some aspects, especially when considering vulnerabilities that emerge from the interaction with external system libraries, these vulnerabilities cannot be prevented through theoretical protocol validation.

We feel that invulnerability might be too strong a notion of security. Besides the practical aspects—it is extraordinarily hard to ensure it—, there are also conceptual reasons. If P belongs to the class of key-exchange (key agreement) protocols, then the security level granted by P is just as high when: (i) P is invulnerable or (ii) honest agents have a way to realize when a key has been compromised and avoid using it.

Exclusive focus on invulnerability has some very concrete consequences on the types of situations classical SPA models can support. As we remarked above, multiple attackers have been a part of SPA just for as long as it took to verify that no new attacks emerge as a consequence of the additional attack power introduced by *cooperation between attackers* (see [18],

[2]Note that in this paper we do not consider general economic approaches based on game theory, e.g., searching for equilibria in the "game" (which in our case would be a protocol under attack). We leave an investigation of the applicability of such approaches for future work.

[19]); from the point of view of attack identification, the interest for multiple attackers ends there.

As a consequence, the standard models in SPA literature assume the presence of a single attacker and are defined accordingly. As we already remarked above, there are repercussions on the communication primitives, which intrinsically reflect the presence of a single attacker and do not enable orderly interaction between multiple attackers. In fact, the set-up is meant to reduce the architecture down to the bare bones, to the point that the network itself is often replaced with the (only) attacker present.

B. Security as compromise with external factors

Some approaches have been proposed to investigate whether it is possible to make safe use of a protocol given the additional knowledge of its specific vulnerability. In a sense, there has been a shift from asking how to run a system with knowledge $\{P\}$ (and how to pick a good P) to asking how to run a system with knowledge $\{P, P_{\text{weaknesses}}\}$, introducing along the way an economic factor capable of acting as a deterrent.

The main example of this shift is the *retaliation* approach, in which an "honest" agent knows it can be attacked, detects security failures after attacks and strikes back against the culprit. In retaliatory settings, honest agents can be treated as full attackers (retaliation through the protocol, if they have knowledge of the attack traces) or as out-of-protocol attackers, in the sense that they exact payment on external resources. The theme is known as "protocol life after attack" [29], [30].

Approaches such as retaliation consider the deployment environment explicitly and note that, in real life, attacking comes with consequences. The attacked tries to discover who has done him wrong, invokes the law, retaliates, calls the police, fights back or retreats. All these mechanisms are quite natural in a scenario in which a group of agents behave according to the rules and some other agent tries to thwart the benefit (i.e., the security property π) that the rules are meant to uphold. In fact, all these mechanisms have been implemented in at least one field of security.

Whereas classical SPA (with collaborative attackers as well) aims at ensuring that all possible weapons against the security protocol are ineffective, retaliation, as all economic approaches, is aimed at developing methods to inhibit the attackers' willingness to carry out attacks.

This approach accepts the existence of effective weapons against the honest agents and attempts to enforce dissuasion—to weaken the attacker's resolve to target an agent. Retaliation moves us away from seeking attack-invulnerability and towards demanding costs for successful attacks. Although costs are to be sustained by attackers, added costs do not protect the system in any way, once the attacker decides to move against the honest agent and to risk paying the associated cost. So, from the perspective of honest agents, the deal is making an attack against oneself a bad solution to the attacker's optimization problem. Clearly, this approach fails completely if the attacker is *not* performing optimization the number or "importance" of the attacks he can carry out.

Economic approaches enforce security as a compromise. The compromise is progressively more in favor of the agent that behaves honestly in P the more this same agent is able to behave as an attacker outside of P. An additional assumption is that the attacker in P is defenseless against retaliation out of P. From the conceptual point of view, this is a world made purely of attackers, partitioned into different domains of activity. Furthermore, defending by turning into an attacker outside of P is subject to the critical flaw of assuming that the attacker also cares about his own state out of P. Retaliating by killing is hardly a defense strategy against a suicide bomber.

Classical SPA agents interact only within a protocol P—and not as full agents but, rather, as roles. In contrast, agents within economic frameworks such as retaliation interact both as classical agents (within P) and as full agents (but only outside of P).

When multiple non-collaborating attackers have been considered in retaliatory settings, the focus of interest was not their behavior within P, but rather it was their behavior with respect to other agents, as in [29], [30], [31], [32]. Here, each protocol participant is allowed to behave maliciously and intercept and forge messages. In fact, each agent may behave as a DY attacker, without colluding nor sharing knowledge with anyone else. The analysis of security protocols under this multi-attacker model allows one to consider scenarios of agents competing with each other for personal profit. Honest agents in this model may also carry out *retaliation attacks*, where an attack is followed by a counterattack, and *anticipation attacks*, where an agent's attack is anticipated, before its termination, by another attack by some other agent.

In contrast to standard DY models, retaliation and anticipation allow protocols to cope with their own vulnerabilities, rather than eradicating them. This is possible because honest agents are capable of doing more than just executing the steps prescribed by a protocol: they can decide to anticipate an attack, or to counter-attack by acting *even after the end of a protocol run* in which they have been attacked.

In a sense, we have already stepped outside of the traditional panorama of SPA, because Alice, Bob and Eve all of a sudden become agents in the sense of AI: they implement strategies and *react* to events. Yet, this innovation stays confined in a post-protocol world: agents retaliate only after an attack has been carried out. In this scenario, multiple attackers that are not reducible to a single one emerge: there is an inherent difference between an attacker that has been discovered, one that has not and one that has suffered a retaliatory attack by an honest agent. Agents are following a more elaborate plot than just the protocol's steps and the attack trace—although the protocol and the attack trace are still important components and are treated as an atomic unit.

The ability to perform retaliation is inextricably intertwined with the problems of (i) detecting an attack and (ii) tracing it to an identity outside of the protocol.

Retaliation tests derive from the protocol, but they have the flavor of meta-reasoning—rather than belonging "inside" the protocol. For example, the challenges in [30] test a property

ϕ such that $\phi \subset$ *semantics*$[P]$. In fact, rather then testing for an *attack behavior*, here honest agents are testing for the correct execution of P. As a consequence, honest agents can retaliate only *after* an attack has succeeded and cannot defend the protocol during the attack itself.

Establishing the target of retaliation is also no easy task, as it requires identifying the agent behind a given role. This necessary component of the "defense" has been implemented in prior studies through explicit identity-revealing challenges [29], [30]; one could also think of implementing ad-hoc reasoning systems, although, in general, solutions have to solve a version of the symbol-grounding problem.

The weaknesses of retaliation can be summarized as follows: "the economic factor is external to P". Payment is exacted only after P has been broken (when P is no longer the current "environment"), within another environment Q (that in which agents who are honest in P can attack) and on the condition that honest agents have managed to solve a difficult identification problem (matching attacker identities in P to targets in Q).

C. Security as compromise within P

Our approach to non-collaborative scenarios considers agents that are interacting in P in a complex manner, not trivially traceable to a predefined attack trace nor reducible to a role of P; competitive attackers make contextual decisions and choose which attack variant to employ on the basis of the messages they have observed in the current run of the protocol. Even honest agents are now *active objects in* P, at the abstraction level at which they are agents and not merely roles.

As we have discussed in Section II, interference between simultaneous attacks naturally results in a multiplicity of outcomes, which are no longer limited just to success (A_P exists and E knows it) or failure (no attack trace is known to attackers). Individual attackers no longer have complete control of the final result of the attack, as their success also depends on factors that are independent of what they do. For example, in [22] we have detailed a case in which P involves a trusted third-party server, which is not constrained to answer requests in any particular order; however, the order matters very much because the first request served will determine which attacker has a chance at succeeding in violating P. Knowledge of the existence of a competitor and knowledge of his identity are further factors that affect the security outcome of an attack.

In such a scenario, a run in which attacker E fails to mount an attack successfully may yield the knowledge necessary to do better in the next round, e.g., by revealing the existence or identity of a competitor. *Interference induces trade-offs*, because the mechanics of interaction under P makes some "failure" states more desirable to an attacker than others.

The fundamental point is that trade-offs are *intrinsic in* P as a non-collaborative execution environment. They are not purposefully introduced by researchers, by stipulating that what occurs during a run of P is related to what happens in some other environment: rather, they are *emergent* as a result of how P itself is structured and governs the admissible interactions. The "economic view" is now applicable directly in P, as opposed to requiring an external world in which agents that are dishonest in P can be dissuaded.

The ability to bring compromise *inside* the execution of P—without having to stipulate anything besides P itself (and the ability of agents to react to their surroundings)—is perhaps the strongest point in favor of treating protocols as environments instead of prescriptions on message exchanges.

The trade-offs induced within P span a diverse range: increased network activity vs. increased chance of being noticed by competitors, chance to succeed/fail in the current run vs. chance to acquire information on competitors, being able to detect one's own failure vs. negating success to competitors. The scenario also opens up the possibility of studying an additional class of tradeoffs in the context of protocols: those related to adaptation, fitness and ecological niches. Relating features of agent configuration to the features of protocols under which the agents perform best would allow laying a foundation of the "ecology" of attacker behaviors.

IV. Conclusions and future work

Computer security revolves around threats, attacks and defenses. One would thus expect that every aspect of security is concerned, in roughly equal measure, with the identification of potential attacks and the development of defenses against such attacks. The field of SPA has, on the contrary, focused almost exclusively on attacks. We have shown that indeed there is room in SPA for a notion of defense: interfering with an attacker *by attacking* creates the necessary room. The conceptual bridge is found in scenarios that involve multiple non-collaborative attackers. Along with the possibility of pursuing defenses for vulnerable protocols, non-collaboration and attack interference also call for a small revolution in system modeling and, especially, in the way we conceptualize protocols.

We contend that the real complexity of P becomes manifest when independent agents interact under it. Protocols can be understood in terms of environments along the three following lines:

1) P describes, norms, structures and regulates the full set of interactions that can occur under it; P induces properties that are relevant to describe the security outcome.
2) Attackers interacting under P face trade-offs that are intrinsic to P; such trade-offs do not require any external factors to manifest.
3) In order for attackers to pursue their own goals effectively, they must have built a prior understanding of the types of situations that can be expected to occur under P; in particular, attackers come to understand that the same state can lead to different security outcomes in a manner that is at least partially independent of their actions. Attackers have to *localize* themselves with respect to the P-worldview on the basis of the messages they can observe.

Taken altogether, these points make a very strong case for protocols *being* environments. With this shift of perspective, it is quite natural to look towards robotics and AI for mature tools, which we plan to recruit to do the following:

- construct an interaction-based picture of P, exploring the worldview of P as attackers would;
- instantiate competitive behaviors for agents whose goals and priorities are known on the basis of the worldview;
- open the path to discover appropriate abstractions on protocol properties, building an understanding of some properties of "protocol runs in execution" that in SPA are currently unexplored.

Abstraction on protocol properties of interest in non-collaborative scenarios can serve as the conceptual basis for a qualitative but rigorous analysis of a given protocol and possibly even serve as a basis for protocol design.

In short, we advocate that the security of protocols should not be evaluated only as to whether the target security property is preserved under an attack, but also according to how much scope protocols provide for honest agents to defend themselves. This stance is related to security in depth: even if attackers succeed in bypassing the protocol's security mechanism (i.e., an attack exists), there might be other mechanisms to put in play to preserve the security property, such as deploying a network guardian.

Along the way, we have also clarified our stance on protocol-life-after-attack: defend, by negating the effectiveness of attacks or lessening their impact. While it is apparent that non-vulnerable protocols are better than vulnerable ones, the ability to defend in the presence of a vulnerability offers us a chance to escape the build-test-and-redo loop that comes with accepting nothing less than invulnerable protocols.

Acknowledgements

The work presented in this paper was partially supported by the FP7-ICT-2009-5 Project no. 257876, "SPaCIoS: Secure Provision and Consumption in the Internet of Services" (www.spacios.eu).

References

[1] The AVANTSSAR Platform for the Automated Validation of Trust and Security of Service-Oriented Architectures, http://www.avantssar.eu.

[2] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. Hankes Drielsma, P.-C. Heám, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò, and L. Vigneron, "The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications," in *Proceedings of CAV'05*, ser. LNCS. Springer, 2005, vol. 3576.

[3] A. Armando and L. Compagna, "SAT-based Model-Checking for Security Protocols Analysis," *International Journal of Information Security*, vol. 6, no. 1, pp. 3–32, 2007.

[4] D. Basin, S. Mödersheim, and L. Viganò, "OFMC: A symbolic model checker for security protocols," *International Journal of Information Security*, vol. 4, no. 3, pp. 181–208, 2005.

[5] B. Blanchet, "An efficient cryptographic protocol verifier based on prolog rules," in *Proceedings of CSFW'01*. IEEE CS Press, 2001, pp. 82–96.

[6] C. Cremers, "The Scyther Tool: Verification, falsification, and analysis of security protocols," in *Proceedings of CAV'08*, ser. LNCS 5123. Springer, 2008, pp. 414–418.

[7] S. Escobar, C. Meadows, and J. Meseguer, "Maude-NPA: Cryptographic Protocol Analysis Modulo Equational Properties," in *FOSAD 2007/2008/2009 Tutorial Lectures*, ser. LNCS 5705. Springer, 2007, pp. 1–50.

[8] G. Lowe, "Casper: a Compiler for the Analysis of Security Protocols," *Journal of Computer Security*, vol. 6, no. 1, pp. 53–84, 1998.

[9] J. Millen and V. Shmatikov, "Constraint solving for bounded-process cryptographic protocol analysis," in *Proceedings of CCS'01*. ACM Press, 2001, pp. 166–175.

[10] S. Mödersheim and L. Viganò, "The Open-Source Fixed-Point Model Checker for Symbolic Analysis of Security Protocols," in *FOSAD 2007/2008/2009 Tutorial Lectures*, ser. LNCS 5705. Springer, 2009, pp. 166–194.

[11] L. Paulson, "The inductive approach to verifying cryptographic protocols," *Journal of Computer Security*, vol. 6, pp. 85–128, 1998.

[12] P. Ryan, S. Schneider, M. Goldsmith, G. Lowe, and B. Roscoe, *Modelling and Analysis of Security Protocols*. Addison Wesley, 2000.

[13] M. Turuani, "The CL-Atse Protocol Analyser," in *Proceedings of RTA'06*, ser. LNCS 4098. Springer, 2006, pp. 277–286.

[14] L. Viganò, "Automated Security Protocol Analysis With the AVISPA Tool," *Electronic Notes in Theoretical Computer Science*, vol. 155, pp. 61–86, 2006.

[15] J. Clark and J. Jacob, "A survey of authentication protocol literature: Version 1.0," 1997.

[16] C. Boyd and A. Mathuria, *Protocols for Authentication and Key Establishment*. Springer, 2003.

[17] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Trans. Inform. Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[18] C. Caleiro, L. Viganò, and D. Basin, "Metareasoning about security protocols using distributed temporal logic," *Electronic Notes in Theoretical Computer Science*, vol. 125, no. 1, pp. 67–89, 2005.

[19] D. Basin, C. Caleiro, J. Ramos, and L. Viganò, "Distributed temporal logic for the analysis of security protocol models," *Theoretical Computer Science*, vol. 412, no. 31, pp. 4007–4043, 2011.

[20] H. Comon-Lundh and V. Cortier, "Security properties: two agents are sufficient," in *Proceedings of ESOP'2003*, ser. LNCS 2618. Springer, 2003, pp. 99–113.

[21] P. Syverson, C. Meadows, and I. Cervesato, "Dolev-Yao is no better than Machiavelli," in *Proceedings of WITS'00*, 2000, pp. 87–92.

[22] M.-C. Fiazza, M. Peroli, and L. Viganò, "Attack Interference in Non-Collaborative Scenarios for Security Protocol Analysis," in *Proceedings of SECRYPT 2011*. SciTePress, 2011, pp. 144–156.

[23] ——, "Attack Interference in Non-Collaborative Scenarios for Security Protocol Analysis [extended version]," May 2011, available at http://arxiv.org/abs/1106.3746.

[24] A. Armando, R. Carbone, L. Compagna, J. Cuéllar, and L. Tobarra Abad, "Formal Analysis of SAML 2.0 Web Browser Single Sign-On: Breaking the SAML-based Single Sign-On for Google Apps," in *Proceedings of FMSE 2008*. ACM Press, 2008.

[25] A. Armando, L. Compagna, R. Carbone, and G. Pellegrino, "Automatic Security Analysis of SAML-based Single Sign-On Protocols," in *Digital Identity and Access Management: Technologies and Frameworks*. IGI Global, 2010.

[26] F. Butler, I. Cervesato, A. Jaggard, and A. Scedrov, "A formal analysis of some properties of Kerberos 5 using MSR," in *Proceedings of CSFW'02*. IEEE CS Press, 2002, pp. 175–190.

[27] D. Basin and C. Cremers, "Degrees of security: Protocol guarantees in the face of compromising adversaries," in *Proceedings of CSL 2010*, ser. LNCS 6247. Springer, 2010, pp. 1–18.

[28] ——, "Modeling and analyzing security in the presence of compromising adversaries," in *Proceedings of ESORICS 2010*, ser. LNCS 6345. Springer, 2010, pp. 340–356.

[29] G. Bella, S. Bistarelli, and F. Massacci, "A protocol's life after attacks," in *Proceedings of 11th International Workshop on Security Protocols*, ser. LNCS 3364. Springer, 2003, pp. 3–18.

[30] ——, "Retaliation against protocol attacks," *Journal of Information Assurance and Security*, vol. 3, pp. 313–325, 2008.

[31] W. Arsac, G. Bella, X. Chantry, and L. Compagna, "Validating Security Protocols under the General Attacker," in *Proceedings of ARSPA-WITS 2009*, ser. LNCS 5511. Springer, 2009, pp. 34–51.

[32] ——, "Multi-attacker protocol validation," *Journal of Automated Reasoning*, vol. 46, no. 3, pp. 353–388, 2011.