A Foul Adversary: Bribery, Extortion and Blackmail (Extended Abstract)

Naipeng Dong and Tim Muller

University of Luxembourg

To analyze security protocols, we need to know the power of the adversary. Usually, it is assumed that the adversary has full control of the Internet. The adversary can also apply cryptographic functions, and even reason about all the information he accumulates in this way. Such an adversary is called a *Dolev-Yao adversary.* Still, we believe there are adversaries that have even more power. The adversary can apply foul play, in particular, the *foul adversary* may bribe or extort participants. He can gain knowledge of their secret information, with a wallet, a gun or a compromising document in his hand.

Bribery, extortion and blackmail are different, not only in effectiveness (as a threat may be more convincing than money), but also in degree of cooperation of the victim. A bribe will usually cause a more active cooperation, whereas a threatened victim may cooperate more reluctantly. However, all various ways of coercion are essentially the same in the perspective of a security analyst: If there is a way for the foul adversary to verify that the victim is cooperating, then the victim must cooperate. If, however, the victim can pretend to cooperate but in reality does not, the victim probably will not cooperate. It does not really matter whether the victim is bribed, extorted or blackmailed.

For example, we have a security property, namely the secrecy of an email e. The sender sends an email $\{e\}_k$, and the receiver is able to open the email with key k. The adversary controls the internet, and can therefore see $\{e\}_k$. He can reason about this message, he can apply keys to it, but it is ineffective. If the foul adversary coerces the sender or the receiver to provide the key, can he can do better? If e is a readable text, then the foul adversary can apply the key k' that was provided by a victim, and decrypt $\{e\}_k$ with it. Only if $k = k'$ will the decryption yield a readable text, so the foul adversary can verify that the victim is cooperating, and hence the victim must cooperate. If e appears like random data, the foul adversary cannot verify whether $k = k'$, and therefore cannot distinguish a cooperating victim from a cheating victim, and the victim thus has no reason to provide the real key k.

Nowadays, the Dolev-Yao adversary is used often in security analysis. It can, in a way, be considered as the strongest adversary, such that, if you are secure against a Dolev-Yao adversary, you are secure against any adversary. It is not, however, obvious what it means to be secure, as it depends on the security property in question. There are simple questions, such as: ‘Does the adversary know my secret information?’ or ‘Does the adversary know who I am?’, but one can compose more complex questions, such as: ‘Even though the adversary does not know who I am, can he figure out that I am the same person as yesterday?’ There is a particular type of compound property, which we will refer to as an enforced property, that states that a property holds under the foul adversary. These enforced properties are a hot topic, due to relevance in emerging areas in security, but also due to the complexity of the compound properties involved. The ability of the adversary to force users is effectively modeled in the security property. Usually this is formulated as the lack of existence of a way for users to cheat the adversary.

In contrast to the classical approach, we propose to strengthen the adversary to a foul adversary, and keeping the security properties simple. This implies that we need to model the ability to force users to provide information as a part of the foul adversaries abilities. As mentioned earlier, the information provided by the user is true, only when the adversary can verify its correctness. Doing so models coercion for information only. However, we are also interested in the foul adversary's ability to force users to perform actions (or make choices, have strategies, etc). We note that, if (and only if) the actions of users in question are private, the foul adversary cannot verify their actions, hence the users will generally not cooperate. The matter of coercing for actions can therefore be reduced to the matter of coercing for information.

There are two reasons to prefer our approach. Firstly, a properly modeled foul adversary can simplify manual and automated security protocol analysis. In the current formalization of security properties, the protocol analyzer needs to explicitly specify a coerced user's behavior in a certain context, which is inherently difficult. Furthermore, checking for the lack of existence of a way for a coerced user to cheat, is a hard problem. Secondly, on a more abstract level, one can wonder where the notion of coercion belongs. Intuitively, it makes more sense to encode coercion as an ability of the adversary, than as a strengthening of a security property.

The method we use to formulate a foul adversary is similar to the method Dolev and Yao used, in the sense that we take the best case scenario for the adversary. If there exists any way for a user A to make the foul adversary know a certain piece of information, then the strategy for A of giving all the information, all the time, is a valid strategy to make the foul adversary know that piece of information. We can turn that statement around, to see that; if A is fully cooperating with the adversary and the foul adversary still does not know a certain piece of information, that information is secret under a foul adversary. Instead of modeling all coerced users sending all pieces of information all the time, it makes more sense to allow the foul adversary to reason using the information. It is, however, important to note that he will still need to verify this information before it becomes his knowledge. Therefore, the foul adversary will learn exactly all verifiable information that the coerced users have. A common way to model the ability of the adversary, is to provide a set of derivation rules, to define the knowledge of the adversary. To model a foul adversary, we take all the derivations of a Dolev-Yao adversary, and add rules for the aforementioned reasoning. The additional reasoning can now be formulated in a single rule; if a coerced agent has information, and the foul adversary can verify this information for correctness, then he has the information too.