Imperfect Information and Intention in Non-Repudiation Protocols

Wojciech Jamroga, Sjouke Mauw, Matthijs Melissen

February 16, 2011

Introduction Repudiation [1] means the *denial* of an entity of having participated in all or part of a communication. When Alice sends a message m to Bob, we can distinguish *non-repudiation of origin* (NRO), which expresses that Alice cannot deny having sent m, and *non-repudiation of receipt* (NRR), which expresses that Bob cannot deny having received m from Alice. Proof of the origin of a message or receipt is provided by digital signatures obtained through a *public key infrastructure*, part of which is a *certificate authority* which links public keys to user identities. An important difference between non-repudiation protocols and most other security protocols is that in non-repudiation protocols, the information that the agents have acquired at the end of the protocol run should not only convince the agent itself, but also serve as a proof towards external agents (such as a judge). Applications of non-repudiation include contract signing and certified e-mail [2]. In this abstract, we show why it is important to take (imperfect) information and intention into account when modelling non-repudiation.

Fair exchange and imperfect information It is often desirable to have a guarantee of fair exchange [3] of non-repudiation. For example, when Alice sends message m to Bob, it should hold that Alice receives her NRR if and only if Bob receives his NRO. One way to formally verify such constraints is by means of ATL [4], a modal logic of strategic ability. In ATL, the formula $\langle\langle A\rangle\rangle \Diamond\varphi$ stands for "Group of agents A has the ability to make sure that at some point in the future φ holds". Kremer & Raskin [1] use the following formula to express one of the conditions of fairness:

$$\Phi \equiv \neg \langle\langle Bob\rangle\rangle \Diamond(\mathsf{NRO} \wedge \neg \langle\langle Alice\rangle\rangle \Diamond \mathsf{NRR})$$

This formula is supposed to express that Bob should not be able to obtain his NRO at some point while at the same time making Alice unable to obtain her NRR. However, in [1], Φ is interpreted in the basic version of ATL which implicitly assumes perfect information. That is, agents are assumed to know precisely the current global state of the system, including the local states of the other agents. Obviously, the assumption is wrong for communication protocols in general (if everybody knows everything, no communication is needed). Moreover, in the specific case of non-repudiation protocols, it can be the case that – even if Alice has a way of behaving that causes her to obtain her NRR – the behaviour cannot be represented as an executable strategy because it requires executing different actions in situations that look the same to her. And even if she has an executable strategy, she may be unaware of having it, and unable to identify it [5]. For example, one can construct a protocol in which Alice needs to send a different message depending on whether Bob did or did not receive some other message. Alice is not aware of messages being received by Bob, so although she has a perfect information strategy, she is not able to follow it.

Conversely, it is also possible that Bob has a perfect information strategy to put Alice at a disadvantage, but the strategy cannot be executed under imperfect information. Thus, it can both happen that a protocol satisfying Φ is clearly unfair, and that a protocol is intuitively fair while satisfying $\neg\Phi$. This problem needs to be addressed by interpreting specifications in the appropriate version of ATL with imperfect information [6].

Intention In many applications, the notion of non-repudiation is not sufficient. Recall that NRO means that the sender cannot deny having sent message m. However, often we require that the sender cannot deny having *intended to* sent m.

When Alice is presented with NRR signed by her, it is clear that she is the origin of the message. However, she can claim that she never executed the protocol in question, and that in fact NRR resulted from another protocol, in which m was used as a random string without meaning. For example, NRR could be interpreted in one protocol as "I am the originator of the message 'I owe you money'", while in a second protocol it is merely used as a confirmation that the participant is still on-line. In this situation, we know that one of the parties cheated: either Bob forced Alice to run the second protocol, or Alice lies about never having executed the first protocol. However, it is not possible to find out which of both parties is the culprit. We call an attack like this a *virtual multi-protocol attack*, as it is not necessary for Alice to actually execute any part of the second protocol; merely the existence of a protocol that could result in the same evidence is sufficient.

In general, for each NRR resulting from a non-repudiation protocol, it is possible to construct another protocol where the message NRR has a different interpretation. Note that adding flags to each message that indicate the purpose of the message does not work, as it is not possible to guarantee that flags are unique across protocols. Note furthermore that the same problem occurs for NRO.

The above problem can be solved by making sure that the certificate authority does not only link public keys and user identities, but additionally stores the exact protocol (and version) for which the key will be used with. Note also that this registration needs to be happening in the physical world, as an electronic registration leads to a bootstrapping problem: the registration authority needs to have non-repudiation of origin, because otherwise the agent whose key is registered could later deny having registered her key.

Conclusion We have studied the security requirement of non-repudiation. Often, fair exchange of non-repudiation needs to be guaranteed. When modelling this property formally, it should be taken into account that agents in a protocol have imperfect information, and therefore a model that takes this into account should be used. Furthermore, we indicate that in many applications the notion of non-repudiation is not sufficient, as the parties are required to send their messages *intentionally*. This can be solved by letting the certificate authority store the exact protocol for which a key will be used.

Acknowledgements This work was supported by the FNR (National Research Fund) Luxembourg under projects S-GAMES, C08/IS/03 and GMASec - PHD/09/082.

References

[1] Kremer, S., Raskin, J.F.: A game-based verification of non-repudiation and fair exchange protocols. Journal of Computer Security **11** (2003)

[2] Dashti, M.T.: Keeping Fairness Alive. PhD thesis, Vrije Universiteit (2008)

[3] Ben-Or, M., Goldreich, O., Micali, S., Rivest, R.: A fair protocol for signing contracts. IEEE Transactions on Information Theory **IT-36** (1990) 40–46

[4] Alur, R., Henzinger, T.A., Kupferman, O.: Alternating-time temporal logic. Journal of the ACM **49** (2002) 672–713

[5] Jamroga, W., van der Hoek, W.: Agents that know how to play. Fundamenta Informaticae **63** (2004) 185–219

[6] Schobbens, P.Y.: Alternating-time logic with imperfect recall. Electronic Notes in Theoretical Computer Science **85** (2004) 82–93