MOVA signature project: University Competition Application

Sebastien Duc

1 Overview

We consider an application where several universities are participating to a tournament. We can imagine that for the first round of the tournament, universities are organized into pairs. Let's consider one pair, and let's denote the two universities by U_1 and U_2. U_1 and U_2 are in competition. In this application, U_1 provides quizzes/challenges/riddles to students in U_2. Similarily U_2 provides quizzes to students in U_1. Students in a university are forming teams. We assume that teams have a certain amount of time (for example one week) to complete the current quiz/challenge/riddle. Each team in a university have a different quiz[1]. When completed they send it to the server of both universities and get a score. At the end of the first round, all the best of the two universities (i.e. the ones with the highest cumulated score of all teams in the university) in the pairs are promoted to the second round. The game goes on the same way until round n with only on pair (only two universities) and the best of the two wins the game.

2 Security

For this part let us consider one pair with universities U_1 and U_2 as before. Let us denote the server of U_1 by S_1 and the server of U_2 by S_2. We can assume that communication between S_1 and S_2 are confidential, autenticated and preserve integrity. When U_1 creates a quiz/challenge for students of U_2 they sign it with MOVA and then send it to them. In that way, the students are ensured that the quiz comes indeed from U_1 and that it is not a fake quiz from other cheating students who wants to make them lose time, hence lose score. Then, when the students are sending back the quiz to S_2, S_2 can sign

[1]If several teams in a university have the same quiz, then they can copy on each other to increase the final score.

the filled quiz with MOVA and send it to S_1. Finally S_1 receives the filled quiz and verify both signatures. For signatures coming from S_2, S_1 can run a batch verification with S_2

3 Problems

First we need to keep authentication and integrity between the students and their university server, because we don't want other malicious students to be able to modify the answer of the quiz. This is not really an issue, we can assume that there is an infrastructure inside the university that allow students and the server to have a secure channel. We also have to assume that S_2 doesn't modify the results of the quizzes before signing them. Again this is not a real issue, because the server is supposed to be honest.

4 Using classical Signatures

In this application we could try to use classical signature. For the signature of the unfilled quiz (the signature signed by S_1, when S_1 is the quiz provider), MOVA does not bring much compared to classical signatures, because

- It is not a problem that the signature is universally verifiable
- We don't need the signature to be invisible
- We cannot use batch verification
- It is not really usefull for the signatures to be very short

For the signature of the filled quiz (signed by S_2, when S_1 is the quiz provider), we exploit the fact that S_1 runs batch verification. But the invisibility property of the signature is useless since communication between the two server is confidential[2].

5 Conclusion

MOVA signatures don't seem to bring much for this application and classical signatures would make things easier. And we need channels keeping confidentiality, authenticity and integrity between the two servers and between the students and the server of their university which should not be a problem in practice.

[2]Maybe even the signature added by S_2 is useless since the channel is confidential.