A 2-Round Anonymous Veto Protocol
(Transcript of Discussion)

Feng Hao

University of Cambridge

The Chancellor is making a speech in the Galactic Security Council, listen up everyone, he says, I propose we should send troops to that enemy planet and occupy it; now is the time for the security council to take a vote.

You are the members of the security council, each one of you has the veto power, and it's time for you to cast your vote. The setting of this problem is that you don't have any private channels, the only way for you to express your opinion is through a public announcement, however, if you publicly declare that you want to veto you would have offended some people, and you will be punished. So the question is, if everything is public, and everything you say, or you send, can be traced back to you as the data origin, how do you send an anonymous message in such an environment. It is mind-boggling this is possible in the first place, but with public key cryptography it is possible. In my talk I am going to present a solution to this puzzle, in addition I will show that the solution is very efficient in almost every aspect.

Frank Stajano: I don't get why you are listing terms with different indices if the summation is over x_i, y_i where the indices are the same?

Reply: No, the y_i expression is in terms of all the x values, *except* x_i. This is the secret aspect of this protocol. The protocol achieves semantic security because, based on the Decision Diffie-Hellman assumption, it is compromised only under full collusion case. In other words, if the Chancellor wants to find out who has vetoed, basically he has to compromise everyone in this room to get the individual private key, which is quite unlikely. And the protocol is resistant to disruptions, so the Chancellor can be involved in the voting as well, but he cannot suppress other people's veto. If he wants to do that, he has to solve the discrete logarithm problem, which is believed to be intractable.

If we compare with the other related techniques, the one technique which is the closest match in terms of performance to ours, is Brandt's protocol. However, the biggest disadvantage in his protocol is that he needs four rounds, but for our protocol we only need two rounds. Why the difference? The difference is because in Brandt's protocol he used the standard El Gamal encryption, the first round is exactly the same as our protocol, and the second round, each participant uses the El Gamal encryption to encrypt the veto message, and it requires two additional rounds to decrypt it securely. But for our protocol, we use some unconventional ways to encrypt the message. We encrypt the message by raising our value to the power of two different values, and the advantage of this approach is that the

B. Christianson et al. (Eds.): Security Protocols 2006, LNCS 5087, pp. 212–214, 2009.

veto message can be decoded immediately after the second round. Also Brandt's protocol has the same system capacity as ours, but because of the difference of constant factors hidden in their O-notation, the actual amount of computation nodes and the bandwidth in his protocol are actually several times more than those in our protocol.

Finally I bring the conclusion. We propose an anonymous veto network, or AV-net, to solve the veto problem, or the dining cryptographers problem. We don't need any secret channels, no third parties, and have no message collisions, it is provably secure on the Decision Diffie-Hellman assumption, and the efficiency is close to the best we can possibly achieve. We think it is quite unlikely there could be any other solutions more efficient that ours. That's all for my talk. Any questions?

James Heather: Have you proof for the claim of maximum efficiency?

Reply: If we look at the protocol, in the first round you only need one exponentiation for each party, that's all, and in the second round, you also need to do just one exponentiation. What less can you get? And for the bandwidth usage, in the first round, the data size you need to send is the underlying group size, and for the second round is also the minimum data size we can possibly achieve. It may be you can get slightly more efficient, but it's quite unlikely. Yes, if you think that is a proof.

Bruce Christianson: If you want to veto, is it important that you raise the x and y_i to some random power, or can I just send a random number?

Reply: No, because you need an honest proof for this. If you send the random numbers, and you are able to suppress the veto, that is why we need an honest proof here.

Bruce Christianson: And so it doesn't matter what order people go in the second round?

Reply: No.

Kenny Paterson: There's another interesting cryptographic primitive called ring signatures[1], have you ever come across this? I guess the issue is that they don't work if you can identify the actual initial sender of the message, you need some kind of mixnet ring signatures that I can place in a computer as coming from me, or from you. If you could put in a ring signature which contains a veto message that could have been signed by any one of a group of people, then it has similar functionalities.

Reply: Yes, I think that is a different underlying assumption, because for this protocol we have very clear authenticated broadcast channel. That means, if you send the message then the people will know exactly it's you who sent the message.

[1] How to leak a secret, Ron Rivest, Adi Shamir, and Yael Tauman, ASIACRYPT 2001, LNCS 2248.

Kenny Paterson: So it's the same solution but with different network assumptions?

Reply: Yes, for the different scenarios, and different cases.

Bruce Christianson: Suppose we're voting on whether to admit someone to our club, and it requires two no votes to blackball, can we generalise this approach in that way?

Reply: In that case I think the general way of doing that is using the secret sharing. You can have, for example, three out of five secret sharing if two people decide not to give any secret sharing, effectively they are doing the vetoing.

Bruce Christianson: But then it's reasonably clear who hasn't given their shares? Which is what you want to avoid.

Reply: Yes, that's true, yes, that's a quite interesting question, I'll look into that.

Craig Gentry: I think you could just do, essentially two of your protocols, and then your proof would be basically that you'd provided proof of knowledge that proves that you didn't do two random values. So you can produce at most one random value, and the other value has to be equal to your x_i, can you do a proof like that efficiently, do you see what I'm saying? In order to accomplish what Bruce wants, you would just basically do a proof of knowledge, in which you would have two values in the group, and you would prove that you have randomised, at most, one of the values, so in other words, umm, actually this doesn't quite work.

Bruce Christianson: Yes, it's a good step though.

Craig Gentry: It's the beginning of a solution, but the problem would be what if the two different objecting people randomised the same thing, one veto would be masked.

Richard Clayton: You can do a slot reservation rather like the dining cryptographers protocol.

Reply: I think that's possible, but on the other hand it is also quite easy to detect. In the full paper we say it's important for the public keys to be all different, you cannot have duplicate. In theory, yes, but in practice, it's just too easy to detect.

Bruce Christianson: Well there's a very small chance that you'll get nobody vetoing by chance in fact.

Reply: You mean the collision? It's extremely unlikely.

Bruce Christianson: Well perhaps we can try and work out how to do a two veto vote.

Reply: Yes, I think that one is future research.