A treatise on the application of Zero Knowledge Proofs

N.P. Smart

No Institute Given

Abstract. This paper discusses some of the security issues related to zero knowledge proofs. We consider a number of problems with these proofs that have not been discussed in the open research community.

1 Introduction

This paper discusses zero-knowledge proofs as they are actually used in the real world, this distinguishes the paper from previous treatments which are mainly theoretical in nature. Recall that a zero knowledge proof is a two party interactive protocol in which one party, the prover, convinces the other party, the verifier, that they know something without the verifier finding out what the prover knows. Such protocols are usually composed either in serial or in parallel to obtain a higher degree of assurance of the final result to the verifier.

A zero knowledge proof usually comes embodied in a larger object which we shall call a paper. The prover, or author of the paper, wishes to convince the verifier or set of verifier's that they have some knowledge. A singular verifier we shall call a referee, whilst a group of verifiers is usually known as a programme committee. The output from the two party protocol is a single bit denoting whether the verifier either accepts or rejects the proof (or paper in which the proof is embedded).

2 Zero-Knowledge Proofs

Zero knowledge proofs in the context that we shall discuss them come in two variants.

Sender Zero-Knowledge

In these the prover convinces the verifier that the prover has zero-knowledge of the subject under discussion. In this situation a zero knowledge protocol will be called sound if the verifier always outputs reject.

Verifier Zero-Knowledge

In these proofs the verifier learns absolutely nothing from the proof, these proofs are said to exhibit perfect zero knowledge and usually terminate with the verifier outputting accept.

As already mentioned a zero knowledge proof is usually accepted via an interactive protocol, in this the prover submits in serial different versions of the proof to the verifier until the verifier either outputs accept or one of the two parties gives up. If the prover gives up then this usually results in the prover accepting that they have zero knowledge and starting to work on a new zero knowledge proof. In the case where the verifier gives up they usually simply output accept. This multi-round protocol should not be confused with the serial composition of protocols in which the prover, after receiving a reject from the verifier, then submits the proof to another verifier (or set of verifier's). This serial composition is encouraged by the standardization body in this area, or IACR as it is commonly known, by ensuring that

- The Crypto submission deadline closely follows the EuroCrypt notification of rejection date.
- The AsiaCrypt submission deadline closely follows the Crypto notification of rejection date.
- The EuroCrypt submission deadline closely follows the AsiaCrypt notification of rejection date.

Another form of composition is that of parallel composition. This is not considered a scientifically sound form of composition, and the standards body dictates that evidence of parallel composition should result in the verifier always outputting reject. It is an open research problem as to whether one can obtain a parallel composition paradigm in which the verifier obtains zero knowledge as to the parallel composition. This higher level form of zero knowledge we shall call meta zero knowledge and is something we shall treat in [1].

One issue that needs to be addressed is how do we know when a protocol exhibits zero knowledge? The standard way this is done in the theoretical literature is by demonstrating a simulation for the proof. The philosophical reasoning is that if the verifier could output a simulation of the protocol to another party without interacting with the prover, i.e. without seeing the paper, then the verifier should clearly be learning nothing. Extensive research has been carried out in the real world and it would appear that deployed zero knowledge proof systems do exhibit this simulation property, in that verifiers comments usually provide no evidence that the verifier has read the proof, or even any of the paper in which the proof is embedded.

One major problem is that of determining whether the verifier is honest or dishonest. There is some evidence that dishonest verifiers (sometimes called friends or supervisors of the prover) exist, this problem is however often mitigated by the use of multiple verifiers (called a programme committee above). However, a major problem still exists in "effectively colluding partially honest verifieras", here a majority of the verifiers output accept or reject simply because the proof of knowledge does not exhibit the properties that the majority expect to occur in the proof. This majority is called "effectively colluding" since it may be that whilst not explicitly colluding they are colluding via a subliminal channel, i.e. they may have all at some point attended the same University, or simply the colluding group do not want to accept yet another paper on stream ciphers.

A more serious problem is one of honest but curious verifiers, these verifiers attempt to learn information from the proof of knowledge protocol and therefore try to break the property we called verifier zero knowledge. Whilst this is a theoretical problem the practical solution is to overload the verifier with a large number of proofs at once. This parallel composition paradigm with respect to the verifier should not be confused with the earlier parallel composition paradigm which was composition with respect to the prover. This defense against honest but curious verifiers opens up interesting research questions in information theory, namely can a large amount of information produce an actual decrease in the information available to the receiving party. This has applications outside that of zero knowledge systems, for example in distributed denial of service attacks.

3 Conclusion

We have discussed the major application of zero knowledge proof systems in the real world. We have also identified a number of open research areas, which will be the subject of a number of zero knowledge embodiments in the coming years.

References

1. N.P. Smart. Meta zero knowledge and the parallel composition paradigm. Simultaneous submission to CRAPTO 2006.