Another Look at "Provable Security". II

Neal Koblitz[1] and Alfred Menezes[2]

[1] Department of Mathematics, University of Washington
koblitz@math.washington.edu
[2] Department of Combinatorics & Optimization, University of Waterloo
ajmeneze@uwaterloo.ca

Abstract. We discuss the question of how to interpret reduction arguments in cryptography. We give some examples to show the subtlety and difficulty of this question.

1 Introduction

Suppose that one wants to have confidence in the security of a certain cryptographic protocol. In the "provable security" paradigm, the ideal situation is that one has a tight reduction (see §4 for a definition and discussion of tightness) from a mathematical problem that is widely believed to be intractable to a successful attack (of a prescribed type) on the protocol. This means that an adversary who can attack the system must also be able to solve the (supposedly intractable) problem in essentially the same amount of time with essentially the same probability of success. Often, however, the best that researchers have been able to achieve falls short of this ideal. Sometimes reductionist security arguments have been found for modified versions of the protocol, but not for the actual protocol that is used in practice; or for a modified version of the type of attack, but not for the security definition that people really want; or based on a somewhat contrived and unnatural modified version of the mathematical problem that is believed to be hard, but not based on the actual problem that has been extensively studied. In other cases, an asymptotic result is known that cannot be applied to specific parameters without further analysis. In still other cases, one has a reduction, but one can show that there cannot be (or is unlikely to be) a tight reduction.

In this paper we give examples that show the subtle questions that arise when interpreting reduction arguments in cryptography.

2 Equivalence But No Reductionist Proof

In [13], Boneh and Venkatesan showed that an efficient reduction from factoring to the RSA problem (the problem of inverting the function $y = x^e \bmod N$) is unlikely to exist. More precisely, they proved that for small encryption exponent e the existence of an efficient "algebraic" reduction would imply that factoring is easy.

R. Barua and T. Lange (Eds.): INDOCRYPT 2006, LNCS 4329, pp. 148–175, 2006.

The paper [13] appeared at a time of intense rivalry between RSA and elliptic curve cryptography (ECC). As enthusiastic advocates of the latter, we were personally delighted to see the Boneh–Venkatesan result, and we welcomed their interpretation of it — that, in the words of their title, "breaking RSA may not be equivalent to factoring" — as another nail in the coffin of RSA.

However, to be honest, another interpretation is at least as plausible. Both factoring and the RSA problem have been studied intensively for many years. In the general case no one has any idea how to solve the RSA problem without factoring the modulus. Just as our experience leads us to believe that factoring (and certain other problems, such as the elliptic curve discrete logarithm problem) are hard, so also we have good reason to believe that, in practice, the RSA problem *is* equivalent to factoring. Thus, an alternative interpretation of the Boneh–Venkatesan result is that it shows the limited value of reduction arguments, and an alternative title of the paper [13] would have been "Absence of a reduction between two problems may not indicate inequivalence."

Which interpretation one prefers is a matter of opinion, and that opinion may be influenced, as in our own case, by one's biases in favor of or against RSA.

3 Results That Point in Opposite Directions

3.1 Reverse Boneh–Venkatesan

A recent result [16] by D. Brown can be seen as giving support to the alternative interpretation of Boneh–Venkaesan that we described at the end of §2. For small encryption exponents e,[1] Brown proves that if there is an efficient program that, given the RSA modulus N, constructs a straight-line program that efficiently solves the RSA problem,[2] then the program can also be used to efficiently factor N. This suggests that for small e the RSA problem may very well be equivalent to factoring. If one believes this interpretation, then one might conclude that small e are more secure than large e. In contrast, the result of Boneh–Venkatesan could be viewed as suggesting that large values of e are more secure than small ones.

As Brown points out in §5 of [16], his result does not actually contradict Boneh–Venkatesan. His reduction of factoring to a straight-line program for finding e-th roots does not satisfy the conditions of the reductions treated in [13]. His use of the e-th root extractor cannot be modeled by an RSA-oracle, as required in [13], because he applies the straight-line program to ring extensions of $\mathbb{Z}/N\mathbb{Z}$.[3]

Brown's choice of title is a helpful one: "Breaking RSA may be as difficult as factoring." All one has to do is put it together in a disjunction with the title of [13], and one has a statement that cannot lead one astray, and accurately summarizes what is known on the subject.

[1] Brown's result actually applies if e just has a small prime factor.

[2] This essentially means that it constructs a polynomial that inverts the encryption function.

[3] For example, when $e = 3$ the polynomial that inverts cube roots is applied to the ring $\mathbb{Z}/N\mathbb{Z}[X]/(X^2 - u)$, where the Jacobi symbol $\left(\frac{u}{N}\right) = -1$.