ANOTHER LOOK AT NON-UNIFORMITY

NEAL KOBLITZ AND ALFRED MENEZES

Abstract. We argue that it is unnatural and undesirable to use the non-uniform model of complexity for practice-oriented security reductions in cryptography.

1. Introduction

In the classical model of computational complexity – also called the *uniform* model – an algorithm, informally speaking, is a fixed set of instructions that can be applied to any admissible input irrespective of its size. One obtains a very different notion – called the *non-uniform* model – if one allows the set of instructions to vary as a function of the input length. One can think of this as an algorithm that for each input length has access to "advice" that helps it solve the problem for any input of the given bitlength. Usually bounds are imposed on the size of the advice strings[1] – for example, the complexity class P/poly consists of all polynomial-time algorithms that are supplied with $\text{poly}(n)$-size advice strings (where n is the input length). But there is no condition on the resources that might be needed in order to generate the advice; for this reason we shall sometimes refer to such algorithms as "unconstructible." An object that is given as part of the advice string must *exist* mathematically, but computationally there need not be any feasible way to find it.

Any uniform algorithm is also a non-uniform algorithm (with empty advice strings), and so a reductionist security proof that is carried out in the uniform model is also valid in the non-uniform model. However, in this paper we shall be interested in proofs that are valid *only* in the non-uniform model. When referring to an algorithm, adversary, construction, proof, or theorem, we shall use the term "non-uniform" to mean that it is valid in the non-uniform model of complexity but *not* in the uniform model.

Raising questions about the usefulness of non-uniform theorems in practice-oriented provable security is controversial. In response to our paper [24], in which we criticized a "coin-fixing" step in the proof of the main theorem of [2] and questioned the value of a theorem that is proved in such a way, Lindell wrote the following comment in the Cryptology ePrint discussion forum (28 February 2012):

Date: June 24, 2012; updated on July 6, 2012.

[1]To avoid trivial situations, we shall always suppose that the advice is required to be of at most subexponential size. If arbitrary exponentially large advice strings were allowed, then the algorithm could be hardwired with an exponential amount of information for each individual input. Considering non-uniform algorithms with restricted exponential bounds on advice size (e.g., $2^{n/3}$, where n is the input length) may lead to interesting questions, but they are outside the scope of this paper.

> There is no flaw whatsoever in the HMAC proof. The so-called flaw pointed out by Koblitz and Menezes is a standard proof in the non-uniform model (where adversaries are modeled as families of polynomial-size circuits, or equivalently as polynomial-time Turing machines with advice). This type of proof is known to anyone who has taken a basic theory of cryptography (or complexity) course...

The purpose of this article is to argue that non-uniform provable security results are very difficult to interpret from a practice-oriented standpoint, are usually far weaker than similar results proved in the uniform model, and in most cases should indeed be regarded as flawed.

Fortunately, to the best of our knowledge the vast majority of security reductions in cryptography are valid in the uniform model. This is true even if one looks at papers by authors who have advocated for non-uniformity and have said that they work in the non-uniform model (an example is the famous paper [4] by Bellare and Rogaway). Thus, we are not claiming that a large number of proofs have to be rejected or rewritten. However, there is reason to be concerned about "non-uniform creep," that is, the propagation of non-uniformity that occurs when authors imitate the proof techniques in another paper or base their proof on results of another paper that uses non-uniform arguments.

Misunderstandings that result from non-uniformity can also propagate when the results of a non-uniform proof are used as a basis for recommendations to practitioners. For example, Rogaway's 2011 evaluation of block ciphers [28] for the Government of Japan contains an erroneous statement of the security assurance provided by the main theorem about NMAC security in [2]. §9.4 of Rogaway's report repeats a fallacious argument of Bellare that would have been valid if his proof had been in the uniform model (see §2 below).

In some cases (e.g., [12][2]) authors might not even have been aware that a cited result is not valid in the uniform model of complexity, and so were not cognizant of the limitations on their own results that follow from their reliance on a paper that used the non-uniform model. Authors who have no intention or desire to weaken their results by using non-uniformity might nevertheless be caught unawares unless they carefully examine all the proofs of cited results to be sure that they are valid in the uniform model. Similarly, practitioners may have to examine an entire proof in order to determine whether an advertised concrete security guarantee is based on the assumption that a certain primitive resists non-uniform adversaries or only uniform ones. Thus, the unclarity in the provable security literature on the use of uniform versus non-uniform models of complexity results in what in our opinion is an unreasonable burden being placed on theoreticians and practitioners alike.

In §2 we present two examples where the non-uniform model led researchers astray. We first recall the flaw in Bellare's analysis in [2] of the concrete security guarantee for NMAC that is provided by the main theorem of that paper. We describe a low-resource adversary that causes the theorem to lose its practical value, and we make a small refinement in our analysis of that adversary's advantage. We then make similar observations about the non-uniform main theorem in the paper [11] by Dziembowski

[2]Personal communication from Pointcheval to the second author, 16 April 2012.

and Pietrzak, which gives a provable security result for their construction of a leakage-resilient stream cipher from a pseudorandom generator. We find that, because of the non-uniformity in their proof, for many practical choices of the parameters the theorem has no content.

With these examples in mind, in §3 we discuss why the different arguments in favor of non-uniformity are unconvincing. In §4 we comment on the tendency to abuse the definition of non-uniform algorithms either by forgetting that the advice string must be the same for all inputs of a given bitlength, or else by considering most of the description of a problem instance not to be part of the input. We also discuss the unfortunate consequences of conflating two very different concepts: (1) auxiliary input in zero-knowledge and (2) non-uniform complexity. In §5 we note the confusing and inconsistent history of non-uniformity in cryptography. In §6 we list some conclusions.

2. Examples

In this section we illustrate the power of non-uniform[3] adversaries by showing how the assumption of security against such adversaries causes the main results of [2] and [11] to largely lose their practical utility. For other questionable uses of non-uniformity, see [3, 32, 33]; however, we shall not analyze the effects of non-uniformity on the theorems in those papers. See [10, 5] for other work showing the strength of non-uniform adversaries.

2.1. **HMAC.** Let $f(K, M)$ be a compression function that maps from $\{0,1\}^c \times \{0,1\}^b$ to $\{0,1\}^c$, where typically the key-length c is 128 (for MD5) or 160 (for SHA1) and the length b of a message block is 512. Let n be a bound on the number of blocks in a message. In [2] Bellare gives a non-uniform proof that essentially says that if f is a pseudorandom function, then so is the message authentication code NMAC (as well as HMAC) constructed from it. In [24] we discussed his use of a "coin-fixing" argument which resulted in an unconstructible f-adversary, and we questioned how one can possibly get a meaningful security bound when a theorem's hypothesis requires resistance to non-uniform adversaries.

In analyzing the practical meaning of his concrete security guarantee, Bellare claims that it "justifies NMAC up to roughly $2^{c/2}/n$ queries." This claim is based on the assumption that there is no generic attack on the pseudorandomness of f that is faster than exhaustive key search. As we commented in [24], this assumption is correct (to the best of current knowledge) in the uniform model, but it is false in the non-uniform one. We first recall our construction of a low-resource unconstructible adversary A that has a much greater advantage than any known uniform adversary in distinguishing $f(K, .)$ (with unknown K) from a random function.

The adversary A is defined as follows. For a non-empty subset S of $\{1, 2, \ldots, c\}$ define $u_S(x)$ for $x \in \{0,1\}^c$ to be the XOR-sum of the subset S of the c bits of x. For any 1-block message M, any S, and any bit t, consider the probability $\text{Prob}(u_S(f(K, M)) = t)$ as K varies over all 2^c keys. Let (M^*, S^*, t^*) be a fixed triple for which this probability is maximal, and hardwire this triple into A. Then A queries M^* to the oracle O; if $u_{S^*}(O(M^*)) = t^*$, it guesses that the oracle is $f(K, .)$, whereas if $u_{S^*}(O(M^*)) = 1 - t^*$, it guesses that the oracle is random. This low-resource adversary has advantage $\geq 2^{-c/2}$

[3] In this section we use the term "non-uniform" in the sense that the word is used by some of the cited authors; see §4 for a discussion of problems with the way the term is frequently used.

unless the compression function f has the property that for every single fixed pair (M, S) (of which there are $\approx 2^{b+c}$) the values $u_S(f(K, M))$ as K varies are much more evenly split between 0 and 1 than would be expected for a random function. The reason for this advantage, as explained in [24], is that as K runs through the 2^c keys, the function $u_S(f(K, M))$ can be modeled by a random walk – the bit t representing a step forward and $1 - t$ representing a step back. The standard deviation of this random walk is the squareroot of the number of steps – that is, $2^{c/2}$ – and so a message M^* that maximizes forward progress is almost certain to take you more than $2^{c/2}$ steps forward from the starting point.

A more refined analysis of the advantage of the adversary A takes into account that, by the Gaussian distribution, a proportion of order $\approx e^{-s^2/2}$ of the triples (M, S, t) is expected to give a walk that ends up at least s standard deviations forward. Setting the reciprocal of this factor equal to the number 2^{b+c+1} of triples and solving for s, we see that a triple (M^*, S^*, t^*) is expected to exist such that the corresponding random walk ends up $\approx \sqrt{b + c}\, 2^{c/2}$ steps forward from the starting point; this gives an advantage of $\sqrt{b + c}\, 2^{-c/2}$.

To be concrete, in [24] we supposed that $n = 2^{20}$, in which case Bellare's claim says that the main theorem of [2] justifies NMAC up to roughly 2^{60} queries for NMAC-SHA1 and 2^{44} queries for NMAC-MD5. We showed that, in view of the adversary A described above, these bounds are wrong, and in fact Bellare's theorem loses content for NMAC-SHA1 if there are 2^{30} queries and for NMAC-MD5 if there are 2^{22} queries. The above refinement of our analysis leads to an additional reduction by a factor of $(b + c)^{1/4}$, which is a little more than 4; thus, the new query bounds for which the theorem loses content are 2^{28} for NMAC-SHA1 and 2^{20} for NMAC-MD5.

The additional factor $(b+c)^{1/4}$ is, of course, small; it is not of earthshaking practical significance. The point of including it in our discussion is not that it's important by itself, but rather that it illustrates how difficult it is to get an accurate estimate of the best possible advantage of any low-resource adversary that *exists* in the mathematical sense. The contrast with the uniform complexity model is striking. As far as uniform algorithms go, the picture is simple: no generic low-resource adversary is known that is more efficient than just trying to guess the hidden key, and that adversary has advantage $\approx 2^{-c}$. (Nor is any low-resource constructible adversary known with advantage $\gg 2^{-c}$ for the specific compression functions used in MD5, SHA1, etc.)

Remark 1. The above discussion used generic adversaries and looked at their effectiveness against compression functions that have good randomness properties. The possibility remains that these adversaries might be even more efficient against a particular compression function. For example, in the case of the MD5 compression function there seems to be no way to rule out the possibility that there may exist a triple (M^*, S^*, t^*) for which the advantage of the adversary constructed above is much greater than $(b + c)^{1/2}2^{-c/2} = 640^{1/2} \cdot 2^{-64}$.

2.2. **Leakage-resilient stream cipher.** A pseudorandom generator $f(K)$ is a function from $\{0, 1\}^c$ to $\{0, 1\}^r$ (where $r \gg c$) that, given a c-bit seed K, produces r output bits that, hopefully, are hard to distinguish from a random sequence unless one knows the seed.

In [11] Dziembowski and Pietrzak give a method for constructing a leakage-resilient stream cipher starting from f. Let n be the number of blocks ("rounds," i.e., calls to f) in their construction, so that their stream cipher has bitlength nr. (We have slightly changed their notation to be consistent with our notation elsewhere.) They give a security reduction that shows that the stream cipher is secure – even if λ bits are leaked in each round – provided that f is secure.

The main theorem in [11] uses the non-uniform model of complexity in order to achieve a reasonably tight security reduction. (See [9] for a discussion of the tightness issue in security proofs.) As the authors explain,

> Throughout the paper, we always consider non-uniform adversaries. In particular, our stream-cipher is secure against non-uniform adversaries, and we require the PRG used in the construction to be secure against non-uniform adversaries. The only step in the security proof where it matters that we are in a non-uniform setting, is in Section 5, where we use a theorem due to Barak et al. [1] which shows that two notions of pseudoentropy (called HILL and metric-type) are equivalent for circuits. In [1] this equivalence is also proved in a uniform setting, and one could use this to get a stream-cipher secure against uniform adversaries from any PRG secure against uniform adversaries. We won't do so, as for one thing the non-uniform setting is the more interesting one, and also the exact security we could get in the uniform setting is much worse (due to the security loss in the reduction from [1] in the uniform setting).

Following the method we used in the case of Bellare's NMAC proof, we now describe a low-resource unconstructible adversary of $f(K)$ that can distinguish between $f(K)$ (with unknown K) and a random bitstring with advantage roughly $\sqrt{r}\, 2^{-c/2}$. Namely, for any subset S of $\{1, \ldots, r\}$ let $u_S : \{0,1\}^r \longrightarrow \{0,1\}$ be the XOR sum of the subset S of the bits. For any subset S and bit t, consider the probability that $u_S(f(K)) = t$ as K varies over all 2^c seeds. Let (S^*, t^*) be a pair for which this probability is maximal, and hardwire this pair into the adversary A. All A does is compute the XOR sum of the subset S^* of the function's output. If this sum is t^*, it guesses that the bitstring is $f(K)$, whereas if it is $1 - t^*$, it guesses that the bitstring is random. The same random walk argument as in §2.1 shows that A has expected advantage $\epsilon_{\text{prg}} \approx \sqrt{r}\, 2^{-c/2}$.

From formulas (4) and (5) in the statement of the main theorem of [11], we find that the bound δ_n on a stream cipher adversary's advantage is greater than $3n^2 2^{\lambda/2} \sqrt{\epsilon_{\text{prg}}}$, where ϵ_{prg} is the advantage of an adversary of the pseudorandom generator. Since the above generic adversary has an advantage of order $\sqrt{r}\, 2^{-c/2}$, we find that

$$\delta_n > n^2 2^{\lambda/2} r^{1/4} 2^{-c/4}.$$

Suppose we choose the following parameters, which are quite reasonable from a real-world standpoint: $n = 2^{12}$, $\lambda = 8$, $r = 2^{16}$, $c = 128$. In that case the bound on the adversary's advantage in the main theorem of [11] is greater than $n^2 2^{\lambda/2} r^{1/4} 2^{-c/4} = 2^{2 \cdot 12 + 8/2 + 16/4 - 32} = 1$; in other words, the theorem has no content. It is interesting to note, however, that if the proof had been valid in the uniform model of complexity, then it would have been reasonable to set $\epsilon_{\text{prg}} = 2^{-c}$, in which case with the above parameter values the adversary's advantage would have been bounded by 2^{-36}.

3. Discussion of the Arguments for Non-uniformity

3.1. **Stronger conclusion, but stronger hypothesis too.** The basic objection to non-uniform algorithms from a practical standpoint is that they work for inputs of length n only if they get some advice a_n *as if by magic*.

Nevertheless, at first glance there are some appealing arguments in favor of non-uniformity. Most obviously, if a security theorem tells us that a protocol cannot be compromised even by non-uniform adversaries, then this conclusion is stronger than resistance just to uniform adversaries. Early in his textbook [14] Goldreich introduces the concept of non-uniform polynomial time. He calls it an "unrealistic" model of efficient computation, but he goes on to say:

> This model will be used only in the negative way, namely, for saying that even such machines cannot do something (specifically, even if the adversary employs such a machine, it cannot cause harm).

Similarly, in [17] Goldwasser argues that

> The most meaningful proofs of security are necessarily those proved with respect to the most powerful adversary. To this end, we should let the polynomial-time adversary be not only probabilistic but also non-uniform (i.e. not fix the algorithm for all input lengths but let the size of the algorithm grow, although bounded by a polynomial function, in the security parameter k).

More recently, in their lecture notes for their 2008 course at MIT [18], Bellare and Goldwasser state:

> Clearly, the non-uniform adversary is stronger than the uniform one. Thus to prove that "something" is "secure" even in presence of a non-uniform adversary is a better result than only proving it is secure in presence of a uniform adversary. (p. 254)

The trouble with this argument for non-uniform proofs is that reductionist security theorems, unlike most theorems in mathematics, are not absolute statements but rather conditional ones. That is, such a theorem needs to assume that a certain mathematical problem is intractable or a certain cryptographic primitive is resistant to adversaries. In the non-uniform model this assumption becomes much stronger than in the uniform model. In other words, the hypothesis as well as the conclusion of the theorem becomes stronger. As we saw in the case of pseudorandomness assumptions on a compression function or pseudorandom generator, a theorem that has such a strong hypothesis is difficult to interpret and, if interpreted carefully, can sometimes be shown to have no practical value.

In practice it is hard to know how to make a concrete evaluation of the intractability of a problem or the advantage of an adversary in the non-uniform model. How can one assess what type of advice strings might exist for a non-uniform adversary? In the uniform case to some extent one can be guided by the state-of-the-art in cryptanalytic algorithms, but that does not help much when the hypothesis of a theorem is non-uniform.

A proof in the uniform model gives the best of both worlds. For example, suppose you have a uniform theorem of the form, "If the RSA problem is intractable, then protocol P

is secure against certain types of adversaries." As we noted before, your uniform proof is also a valid proof in the non-uniform model. Thus, you really get two theorems for the price of one: (1) "if the RSA problem cannot be solved by a uniform algorithm, then protocol P cannot be attacked by a uniform adversary;" and (2) "if the RSA problem cannot be solved even by a non-uniform algorithm, then protocol P cannot be attacked even by a non-uniform adversary."

3.2. **Does concreteness suggest non-uniformity?** Another argument in favor of non-uniform proofs is that in practice one is interested only in a certain range of input lengths. No one needs to factor 50,000-bit RSA moduli or find discrete logarithms on an elliptic curve over a 1000-bit prime field. So why insist that a single circuit handle *all* possible input lengths? The above passage from Goldwasser [17] continues as follows:

> The latter [a non-uniform adversary] is appropriate since cryptosystems are often designed with a fixed size (that is, security parameter) in mind.

However, that argument is unconvincing because the notion of an algorithm that works efficiently only for inputs in the practical range is not the same thing as a non-uniform algorithm. The latter concept is both stronger and weaker. On the one hand, the statement that an algorithm is polynomial time in the non-uniform model says nothing about whether or not the advice strings can be feasibly constructed, and even polynomial-size advice may take exponential time to find. Hence, having a non-uniform algorithm can fall far short of having a practical algorithm. On the other hand, an algorithm that applies only to a limited range of input lengths is much less than a non-uniform algorithm, since for the latter one needs to have circuits for all possible input lengths. Thus, non-uniformity is a very different concept from practicality.

In informal discourse a word that has a precise definition might be used in a colloquial sense that is quite different from – even inconsistent with – its technical meaning. Suppose one is interested in a problem with a fixed input length or even a fixed input – for example, breaking pseudorandomness of the AES256 compression function. The distinction between uniformity and non-uniformity does not apply here, since every algorithm is both uniform and non-uniform if there is only one permissible input length. Nevertheless, it might be of interest to ask about the effect on running time of allowing an advice string with a specific bound on its bitlength. Such questions can be said to be in the "spirit" of non-uniformity, informally speaking.

More generally, the study of algorithms that need massive precomputation may be regarded as somewhat analogous to non-uniform complexity. The analogy is loose and imprecise, in part because in the former study one has to be concerned with the running time of the precomputation; in fact, it can be argued that the concrete study of precomputation has more in common with uniform complexity, where all algorithms are constructible, than with non-uniform complexity, where there might be no feasible way known to find the advice strings. In any case, whatever meaning one might give the term "non-uniform" in informal speech, such usage cannot serve as justification for giving formal reductionist security proofs in the non-uniform model of complexity, where any result has to be based on the assumption that a mathematical problem is intractible or a cryptographic primitive is secure even in the face of unconstructible algorithms and adversaries.

3.3. **Auxiliary information.** A third argument in support of non-uniformity that one sometimes hears is the following. An adversary often has a certain amount of possibly helpful information when a protocol begins, and a definition should allow an adversary to possess such information during its attack on the protocol. On p. 495 of [15], writing about zero-knowledge, Goldreich says that one needs a definition that will "account for auxiliary information that an adversary B^* may have prior to entering the interaction." Also writing about zero-knowledge definitions, Goldwasser, Micali, and Rackoff [20] give the following explanation:

> Second, it should be explained why B' sees an additional string H.... H may be thought about in a number of different ways; H may be some extra information that the verifier (cheating or not) happens to know. For example, a zero-knowledge protocol for graph isomorphism should remain zero-knowledge even if the verifier happens to know colorings for the graphs. It is also possible that the protocol will be inserted in the middle of another protocol, where the verifier has seen some history H. There is the fear that this H was generated perhaps by interacting with a machine of unlimited power; we want to rule out the possibility of the verifier then obtaining knowledge by using H when interacting with A.

These observations about allowing for auxiliary input have often been interpreted as calling for the non-uniform model of complexity. For example, in the Crypto 2007 paper [30] we see the phrase "*with respect to auxiliary input* (or equivalently, with respect to non-uniform adversaries)" (p. 206; italics in original). Later on the same page Unruh makes the sweeping claim that

> ...most modern cryptographic schemes are designed to be secure even in the presence of an auxiliary input (given that the underlying complexity assumptions hold against non-uniform adversaries).

In other words, one should use non-uniform complexity in order to take account of auxiliary information.

However, in cryptographic practice the most common sources of auxiliary information that might threaten the security of a protocol are: (1) earlier runs of the protocol or of different protocols with which it is composed or in which it is embedded, (2) massive precomputation, and (3) side-channel attacks. In most cases non-uniform complexity does a poor job of modeling the adversary's capabilities. This is because usually the information that is generated by (1), (2) or (3) is not the same for all inputs of a given bitlength, but rather depends on the particular parameter values that are being used.[4] Thus, a result that says that a given protocol is secure against non-uniform adversaries does not provide any assurance at all about what happens when the protocol is confronted by an adversary that generates information in any of the ways (1)-(3) that typically occur in practice.

3.4. **Zero-knowledge.** A related argument in favor of non-uniformity is that the non-uniform model of complexity is basic to the theory of zero-knowledge, and for this reason one should expect it to be used in other areas of cryptography as well. After the first

[4]An exception in the case of integer factorization is given in [5], where the massive precomputation depends only on the bitlength of the number being factored.

version of our paper [24] was posted, on 23 February 2012 Bellare sent an email to us that concluded with the following paragraph:

> You may also want to reflect that non-uniform complexity is used widely in other parts of cryptography. For example, the classical treatment of zero-knowledge uses it. Goldreich's paper on a uniform treatment moves to the uniform case but the bulk of work still uses non-uniform.

In this subsection, after recalling the basic setting in zero-knowledge, we describe what is actually a very limited role of non-uniform complexity in that theory.

Non-uniformity may arise when proving that a certain interaction reveals no information other than what it was designed to. This is the *zero-knowledge* property, not to be confused with *soundness* (the assertion that the prover Alice cannot succeed unless she really knows the secret information) and *completeness* (the assertion that Alice will be able to fulfill the demands of the verifier Victor if she does know the secret). While soundness is normally demonstrated by a reduction argument (and the proof of completeness is usually very easy and direct), the zero-knowledge property is proved by means of a simulation argument – which is very different from giving a reductionist security proof for a protocol. That is, to prove zero-knowledge one constructs a simulator algorithm that for all but finitely many inputs creates a view that one shows is indistinguishable from the view seen by Victor in his interaction with Alice. Since the simulator does not interact with Alice or have access to any secret knowledge she has, it follows that Victor's interaction with Alice could not have leaked any information.

Notice that the zero-knowledge property is independent of whether or not the protocol is secure in the sense of the soundness property. Suppose, for example, that Alice has an RSA public modulus N, and is using a zero-knowledge proof of her knowledge of the factors of N in order to prove her identity to Victor. The statement that the protocol has the zero-knowledge property means, informally speaking, that there is no way that Victor can use his interaction with Alice to get any knowledge that will make it easier for him to factor N than it was before the interaction; the protocol has the auxiliary-input zero-knowledge property if this remains true no matter what additional information Victor has when the interaction begins. It remains true even if Victor has already learned the factorization of N (although what it says in that case is trivial).

Let Dick the Distinguisher be an adversary of the proof by simulation of the zero-knowledge property. Most commonly that is interpreted to mean that there exists an infinite sequence of inputs corresponding to increasing security parameters such that Dick, with the possible help of auxiliary input, is able to distinguish between Victor's view and the simulator's view. In this interpretation Dick is not required to find the sequence of inputs; rather, in each case Dick has an advice string consisting of the input together with the corresponding auxiliary input. Thus, Dick is a non-uniform algorithm.

However, in [13] Goldreich shows that the theory can be developed using an alternative interpretation, where Dick is required to find the inputs for which he can distinguish Victor's view from the simulator's view. In other words, Dick is now a uniform algorithm (but with auxiliary input). In [13] Goldreich also warns against the mistaken notion that one can take care of the need for auxiliary input by using the non-uniform model of complexity: "Hence, using a non-uniform formulation without introducing an auxiliary input to the machines (and in particular to B^*) is not satisfactory" (p. 38). He

repeats this warning in [14] in a section titled "Why Not Go for a Fully Non-Uniform Formulation?," where he says that "the impression that non-uniform circuits account for auxiliary inputs is wrong."

In that section Goldreich mentions another difficulty with the "fully non-uniform formulation," namely, that it does not give the desired notion of zero-knowledge. We can illustrate this using an example where there are only a few permissible inputs for each input length. The Cunningham Project (see [6]), which played an important role in the development of computational techniques in number theory, was concerned with finding the complete factorization only of numbers of the form $b^n \pm 1$ for $b \leq 12$ and large n. Alice's zero-knowledge protocol for proving that she knows the factorization of N should remain meaningful if one restricts to N of the form $b^n \pm 1$, $b \leq 12$. However, in the fully non-uniform model zero-knowledge has no content in that setting, because a non-uniform verifier Victor can trivially factor the sequence $b^n \pm 1$.

Thus, following Goldreich [13, 14], we conclude that non-uniform complexity has at most a limited role in zero-knowledge – namely, in reference to Dick the Distinguisher – and even there one can just as well use uniform algorithms with auxiliary input. Moreover, the concepts of auxiliary input and non-uniformity are quite different and should not be conflated.

Note that the soundness property of a zero-knowledge protocol – which is the only one of the three properties that is proved using a reduction – is, as Goldreich points out, best proved in the uniform model of complexity. There is no logical basis for citing zero-knowledge proof methodologies as a reason for advocating non-uniformity in security reductions.

4. Misusing the definition of non-uniformity

If the notion of a non-uniform algorithm is used correctly, then in many contexts it makes little difference if one asks for a uniform or non-uniform algorithm. For example, in the case of the integer factorization problem there is no known non-uniform algorithm with polynomial-size advice strings that is faster than all known uniform algorithms.[5] One can imagine polynomial-size advice a_n that could help factor certain classes of n-bit integers, but it is hard to conceive of such advice that would speed up the factorization of *arbitrary* n-bit integers. In his textbook [14] Goldreich emphasizes this point:

> We stress that [the non-uniform] machine M gets the same advice (i.e., a_n) on all inputs of the same length (i.e., n). Intuitively, the advice a_n may be useful in some cases (i.e., for some computations on inputs of length n), but it is unlikely to encode enough information to be useful for all 2^n possible inputs.

4.1. **Discrete log problems.** When one speaks of the input to a computational problem, one normally means *all* the information needed to describe the problem instance. Even if certain parameters for a protocol are fixed system-wide, that does not mean that they are not part of the input. For example, the input to a discrete log problem includes a description of the group (such as the size q of the finite field or the equation of the elliptic or hyperelliptic curve), even if this group is fixed in a given implementation

[5] If one relaxes "polynomial-size" to "subexponential-size," then this is no longer true, according to a recent result of Bernstein and Lange [5].

of the protocol. By definition, the advice supplied to a non-uniform algorithm must be the same for all inputs of a given total bitlength.

Remark 2. For certain restricted problems there might be only one or a small number of groups allowed for each input length; then the advice can depend on the particular group. This would be the case, for example, if one is interested in solving the discrete log problem only in prime fields of Mersenne-prime size or on anomalous binary elliptic curves (of which there are only two for each extension field $\mathbb{F}_{2^n}$ if n is odd and one if n is even).

A similar comment applies to the integer factorization problem. By definition, a non-uniform algorithm for the Cunningham Project (see §3.4) would be permitted to have different advice strings for each permissible input. (In the non-uniform model of course the Cunningham Project is trivial.)

Remark 3. Even if a proposed algorithm does not satisfy the definition of a non-uniform algorithm, it might nevertheless be a worthwhile contribution to computational mathematics. For example, it would be of great interest to solve any of the following open problems, although in no case would such a solution give a non-uniform algorithm:

Open problems. (1) Give a polynomial-time algorithm that, given any finite field $\mathbb{F}_q$ and a polynomial-size advice string that depends only on q, computes discrete logarithms in $\mathbb{F}_q$.

(2) Give a polynomial-time algorithm that, given any elliptic curve E defined over a finite field and a polynomial-size advice string that depends only on E, computes discrete logarithms in E.

(3) The same as (2) with "polynomial" replaced by "subexponential," or by $N^{1/4}$, where N is the order of the group E.

In this section when we ask whether or not a proposed construction satisfies the definition of a non-uniform algorithm, we are not addressing the separate question of whether or not it is an interesting or worthwhile construction.

An erroneous use of the term "non-uniform" occurred in the work [26, 27] of Maurer and Wolf, who claimed to have constructed (under a certain smoothness assumption about numbers in the Hasse interval) a non-uniform reduction from the discrete log problem to the Diffie-Hellman problem. The theorem in a subsection of [27] titled "Non-uniform equivalence" starts out as follows:

> Let P be a fixed polynomial. For every finite cyclic group G with order $|G| = \prod p_i^{e_i}$ and such that all multiple prime factors p_i of $|G|$ are smaller than $P(\log |G|)$, there exists an algorithm ...

Notice that the discrete log and Diffie-Hellman problems being considered allow exponentially many possible groups for each value of $\log |G|$ and hence for each input length. The authors then explain that "for every cyclic group G there exists a small piece of information, which depends only on the order of G, that makes breaking the Diffie-Hellman protocol and computing discrete logarithms equivalent in G." Here the advice string is much more than what's permitted in a non-uniform algorithm, because it is not the same for all group orders of a fixed bitlength.

Remark 4. The series of papers by Maurer and Wolf on the discrete log and Diffie-Hellman problems went a long way toward establishing the equivalence of these two

fundamental problems. Our purpose in raising the non-uniformity issue here is not in any way to question the value of their work, but rather simply to point out an error in terminology. Is it nit-picking pedantry to do this? Perhaps. However, researchers who believe in the importance of theorem-proving for establishing the security of protocols frequently argue that one of the advantages of a rigorous mathematical approach is that it forces cryptographers to be careful and precise in their terminology. Indeed, in [22] Katz and Lindell give "formulations of exact definitions" as Principle 1 of the "basic principles of modern cryptography." While terminological imprecision might not be worthy of note among people who use *ad hoc* engineering approaches to evaluate the security of protocols, it is disconcerting to find striking cases of inaccurate word usage – for instance, a fundamental notion such as "non-uniform" used incorrectly – in the provable security literature. See [23] for more discussion of the value of precise definitions.

4.2. **Secret sharing.** A more recent example of the same misunderstanding of non-uniform complexity can be found in the Crypto 2011 paper [8], which constructs an arithmetic secret sharing scheme over a finite field $\mathbb{F}_q$ that has certain strong properties. It turns out that in order to construct the scheme one needs to be given an advice string that no one knows any feasible way to find. As the authors explain, their general construction is not valid in the uniform model of complexity:

> As to efficiency, when given a divisor that is a solution to these Riemann-Roch systems, it is efficient to compute a generator for the scheme C defined by the divisor. However, solving Riemann-Roch systems efficiently in full generality is subject of further research. In particular, for our strongest results to follow it is not known at present how to efficiently compute a generator. But of course, there exists a poly(n)-size description of generators, so overall, there is efficiency as before, but now in the weaker model where such description is given as advice.

Despite the impression given by the last sentence of this paragraph, the construction in [8] is not valid in the non-uniform model either, since the advice string depends on the field $\mathbb{F}_q$, and so is not the same for all inputs of a given bitlength.

Remark 5. We are not questioning the validity of the theorems in [8], which are statements about the mathematical existence of certain secret-sharing schemes. Nor are we questioning the correctness of the security claims in [8] about such schemes. Rather, the issue here is the actual construction of the schemes.

4.3. **Symmetric key.** A somewhat different type of abuse of non-uniformity occurs in symmetric-key cryptography. Rigging a computational problem so that there is essentially just one input for each input length seems to us to result in a questionable use of the notion of a non-uniform algorithm. For example, suppose that someone is proving a theorem about message authentication codes that are constructed from compression functions f. The security reduction assumes one has a MAC adversary and uses it to construct an f-adversary. The proof is non-uniform, and so the latter adversary can have some advice wired into it. Now the question is: Can that advice depend upon the function f? For instance, can it consist of a message or pair of messages that have some unusual property with respect to the particular function f? Or must it be the same

advice if f is changed to a different f^* whose description has the same bitlength? For instance, suppose that the description of f involves some fixed vectors or S-boxes. One could change the fixed vectors without changing the bitlength of the description of the compression function.

A troubling feature of proofs such as the NMAC proof in [2] is that the specification of the function is not considered to be part of the input, but rather is regarded as fixed (more precisely, such theorems assume a fixed family of compression functions, one for each security parameter). Through this sleight of hand one gets the right to give the algorithm advice that depends on the specific compression function. This seems to be a dubious use of the notion of a non-uniform algorithm.

Notice that in settings where the adversary's task is an NP-search problem, no one would be so foolish as to give a non-uniform proof with fixed functions, since such a proof would be trivial and empty. For example, suppose that a theorem assumes collision-resistance of a hash function. Such a theorem would be a useless and misleading triviality if one regarded the hash function as fixed and allowed non-uniform adversaries (which could then have collisions wired into them).

4.4. **Confusing auxiliary input with non-uniformity.** As we saw in §3.3, experts in zero-knowledge have for a long time insisted on the need to take account of auxiliary input in the definitions. Summarizing the consensus, Goldwasser and Kalai [19] say simply that "by now, the terms zero-knowledge [21] and auxiliary-input zero-knowledge [16] have become one and the same." We also saw that, unfortunately, some writers interpret this as an argument for non-uniform complexity. There are several reasons why this is a bad idea.

First, as we explained in §3.3, auxiliary input almost never is the same for all problem inputs of a given bitlength, and so the term "non-uniform" is incorrect. We find it perplexing that people who believe in the importance of mathematical rigor in cryptography would misuse fundamental terminology. When one uses "non-uniform" to also mean "auxiliary-input," one is making the word work double-shift.

> 'That's a great deal to make one word mean,' Alice said in a thoughtful tone.
>
> 'When I make a word do a lot of work like that," said Humpty Dumpty, 'I always pay it extra.' ([7], p. 191)

Second, the use of the wrong term causes some people to conclude that, if the non-uniform model is pervasive in the theory of zero-knowledge, then it must be appropriate to use it elsewhere in cryptography, in particular, in reductionist security arguments. If, on the other hand, one understands that auxiliary input means something different from non-uniformity, then one will not be led astray by this specious argument.

Third, the confusion between the two terms could lead someone to believe that a suite of protocols has a fundamental security weakness when in fact it does not. Namely, suppose that one of the protocols (say, for proof of identity) is shown to have the zero-knowledge property with auxiliary input, and we mistakenly believe that this means that the non-uniform model of complexity is being used in the reductionist arguments for the soundness of the proof of identity and for the security of other protocols in the suite. That means that any reductionist security theorem must be interpreted in the non-uniform model, and this might greatly affect what such a theorem says about

concrete security. Thus, we could end up with a severe *under*-estimate of the security of the suite of protocols.

Some cryptographers apparently believe that, in the words quoted earlier from [30], "most modern cryptographic schemes are designed to be secure even...against non-uniform adversaries." This is, in fact, false. And, fortunately, there is no practical need for most protocols to be secure against non-uniform adversaries. If that type of security were necessary, then many protocols at the very least would have to greatly increase their parameter sizes and thus lose efficiency.

5. History

A half-century ago in a paper [31] titled "The unreasonable effectiveness of mathematics in the natural sciences" the physicist Eugene Wigner famously wrote that

> ...the enormous usefulness of mathematics in the natural sciences is something bordering on the mysterious and...there is no rational explanation for it.

In the computational sciences, an example of the surprising practicality of a purely mathematical notion is found in the concept of polynomial-time algorithms. There is no *a priori* reason why this term should be taken to imply "feasible." After all, an algorithm with running time $O(n^c)$ (where n denotes input length) is not very practical if $c = 50$. However, the empirical fact is that virtually all polynomial-time algorithms that arise in practical problems of any importance are practical algorithms.

During this year of celebration of the centenary of Alan Turing's birth, it is worthwhile to recall that another example of the unreasonable effectiveness of mathematics is given by the model of computation that Turing defined in [29]. Despite many decades of intensive development of algorithms and despite the increase in computing power between 1936 and 2012 by many orders of magnitude, the notion of a Turing machine remains central to any systematic discussion of algorithms. This notion is, of course, the *uniform* model of complexity.

In cryptography, Turing's approach to computational complexity has been challenged by researchers who claim that non-uniform complexity is better suited to concrete analysis of protocols. This claim – which we believe to be deeply flawed – has had a rather confusing history.

On the one hand, twenty years ago one of the leading theoreticians gave a clear explanation of the drawbacks of non-uniform complexity. In his seminal article [13], Goldreich wrote:

> A uniform transformation [security reduction] implies a non-uniform one, whereas the converse is not always true. Hence, the uniform transformation supplied by the uniform treatment is technically superior to the non-uniform one.... Another reason to prefer the uniform treatment is that it is based only on uniform intractability assumptions. These assumptions are seemingly weaker (and never stronger) than their non-uniform counterparts....
>
> However, something is lost when relying on these (seemingly weaker) uniform assumptions. Namely, the security we obtain is only against the (seemingly weaker) uniform adversaries. We believe that this loss

> in security is immaterial. Our belief is based on the thesis that uniform complexity is the right model of "real-world" cryptography. We believe that it is reasonable to consider only objects (i.e., inputs) generated by uniform and efficient procedures and the effect that these objects have on uniform and efficient observers (i.e., adversaries).

In the same vein, in his book [25] Luby wrote: "In general, a uniform reduction is much more desirable than a non-uniform reduction" (p. 34).

At the opposite extreme, some prominent researchers in practice-oriented provable security have adamantly maintained that the non-uniform model of complexity is preferable to the uniform model. In a 24 February 2012 email to the first author, who had asked why the paper [2] nowhere mentions that the main theorem is valid only in the non-uniform complexity model, Bellare wrote the following:

> My paper uses a concrete complexity framework. Such a framework is inherently non-uniform. This has been understood since such frameworks started.... [W]hen complexity is concrete, we have non-uniformity.... I had no idea my paper would be read by anyone not familiar with the fact that concrete security is non-uniform.

6. Conclusions

1. Uniform proofs are preferable to non-uniform ones. Non-uniform security reductions should be avoided if possible.

2. Terms such as "non-uniform complexity" that have precise meanings should be used correctly so as to avoid misunderstandings. The notion of auxiliary input should never be confused with the concept of a non-uniform algorithm or proof.

3. If a paper contains a non-uniform security reduction, the introduction should give the reader a clearly worded warning about this.

4. A non-uniform reductionist security theorem should contain a clear and complete statement of the input to the intractable problem or adversarial task in the hypothesis, so that a reader understands what it means to allow advice strings that must be the same for all inputs of a given bitlength.

5. Authors who prove security results in the non-uniform model should be very cautious about claiming any concrete security guarantee at all.

Acknowledgments

This paper was inspired by a suggestion of Mihir Bellare, who on 23 February 2012 emailed us that "you are welcome to critique non-uniform complexity to your heart's content." We thank him for this suggestion.

We wish to thank Paul Beame, Joan Feigenbaum, Mike Fellows, Steven Galbraith, Neil Immerman, Rahul Jain, and Ashwin Nayak for helpful discussions of non-uniform complexity. Of course, none of them is responsible for any of the opinions expressed in this article.

We also thank Ann Hibner Koblitz for editorial corrections and suggestions.

References

[1] B. Barak, R. Shaltiel, and A. Wigderson, Computational analogues of entropy, *7th International Workshop on Randomization and Approximation Techniques in Computer Science*, 2003, pp. 200-215.

[2] M. Bellare, New proofs for NMAC and HMAC: Security without collision-resistance, *Advances in Cryptology – Crypto '06*, LNCS 4117, Springer-Verlag, 2006, pp. 602-619; extended version available at http://cseweb.ucsd.edu/mihir/papers/hmac-new.pdf

[3] M. Bellare and T. Ristenpart, Multi-property-preserving hash domain extension and the EMD transform, *Advances in Cryptology – Asiacrypt 2006*, LNCS 4824, Springer-Verlag, 2006, pp. 299-314.

[4] M. Bellare and P. Rogaway, Random oracles are practical: A paradigm for designing efficient protocols, *First ACM Conference on Computer and Communications Security*, ACM, 1993, pp. 62-73.

[5] D. Bernstein and T. Lange, Non-uniform cracks in the concrete: the power of free precomputation, available at http://cr.yp.to/nonuniform/nonuniform-20120604.pdf

[6] J. Brillhart, D. Lehmer, J. Selfridge, B. Tuckerman, and S. Wagstaff, *Factorizations of* $b^n \pm 1$, $b = 2, 3, 5, 6, 7, 10, 11, 12$, *Up to High Powers*, AMS, 2002.

[7] L. Carroll, *Alice's Adventures in Wonderland* and *Through the Looking-Glass*, Oxford Univ. Press, 1971.

[8] I. Cascudo, R. Cramer, and C. Xing, The torsion-limit for algebraic function fields and its application to arithmetic secret sharing, *Advances in Cryptology – Crypto 2011*, LNCS 6841, Springer-Verlag, 2011, pp. 685-705.

[9] S. Chatterjee, A. Menezes and P. Sarkar, Another look at tightness, *Selected Areas in Cryptography — SAC 2011*, LNCS 7118, Springer-Verlag, 2012, pp. 293-319.

[10] A. De, L. Trevisan, and M. Tulsiani, Non-uniform attacks against one-way functions and PRGs, *Electronic Colloquium on Computational Complexity*, TR09-113, 2009.

[11] S. Dziembowski and K. Pietrzak, Leakage-resilient cryptography, *Symposium on the Foundations of Computer Science*, IEEE, 2008, pp. 293-302.

[12] P. Fouque, D. Pointcheval, and S. Zimmer, HMAC is a randomness extractor and applications to TLS, *Symposium on Information, Computer and Communications Security – AsiaCCS 2008*, ACM Press, 2008, pp. 21-32.

[13] O. Goldreich, A uniform-complexity treatment of encryption and zero-knowledge, *J. Cryptology*, **6** (1993), pp. 21-53.

[14] O. Goldreich, *Foundations of Cryptography*, Vol. 1, Cambridge Univ. Press, 2001.

[15] O. Goldreich, *Computational Complexity: A Conceptual Perspective*, Cambridge Univ. Press, 2008.

[16] O. Goldreich and Y. Oren, Definitions and properties of zero-knowledge proof systems, *J. Cryptology*, **7** (1994), pp. 1-32.

[17] S. Goldwasser, The search for provably secure cryptosystems, in C. Pomerance, editor, *Cryptology and Computational Number Theory*, *Proc. Symposia in Applied Math.*, **42** (1990), pp. 89-113.

[18] S. Goldwasser and M. Bellare, Lecture Notes on Cryptography, July 2008, available at http://cseweb.ucsd.edu/~mihir/papers/gb.pdf

[19] S. Goldwasser and Y. Kalai, On the impossibility of obfuscation with auxiliary input, *Symposium on the Foundations of Computer Science*, IEEE, 2005, pp. 553-562.

[20] S. Goldwasser, S. Micali, and C. Rackoff, The knowledge complexity of interactive proof systems, *SIAM J. Computing*, **18** (1989), pp. 186-208.

[21] S. Goldwasser, S. Micali, and R. Rivest, A digital signature scheme secure against adaptive chosen-message attacks, *SIAM J. Computing*, **17** (1988), pp. 281-308.

[22] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, Chapman and Hall/ CRC, 2007.

[23] N. Koblitz and A. Menezes, Another look at security definitions, available at http://anotherlook.ca

[24] N. Koblitz and A. Menezes, Another look at HMAC, available at http://anotherlook.ca

[25] M. Luby, *Pseudorandomness and Cryptographic Applications*, Princeton Univ. Press, 1996.

[26] U. Maurer and S. Wolf, The relationship between breaking the Diffie-Hellman protocol and computing discrete logarithms, *SIAM J. Computing*, **28** (1999), pp. 1689-1721.

[27] U. Maurer and S. Wolf, The Diffie-Hellman protocol, *Designs, Codes, and Cryptography*, **19** (2000), pp. 147-171.
[28] P. Rogaway, Evaluation of some blockcipher modes of operation: Evaluation carried out for the Cryptography Research and Evaluation Committees (CRYPTREC) for the Government of Japan, 10 February 2011, available at http://www.cs.ucdavis.edu/~rogaway/papers/modes-cryptrec.pdf
[29] A. M. Turing, On computable numbers, with an application to the Entscheidungsproblem, *Proc. London Math. Soc.* (2), **42** (1936), 230-265; and **43** (1937), 544-546.
[30] D. Unruh, Random oracles and auxiliary input, *Advances in Cryptology – Crypto 2007*, LNCS 4622, Springer-Verlag, 2007, pp. 205-223.
[31] E. P. Wigner, The unreasonable effectiveness of mathematics in the natural sciences, *Communications in Pure and Applied Math.*, **13** (1960), 1-14.
[32] K. Yasuda, "Sandwich" is indeed secure: How to authenticate a message with just one hashing, *Information Security and Privacy – ACISP 2007*, LNCS 4586, Springer-Verlag, 2007, pp. 355-369.
[33] K. Yasuda, Boosting Merkle-Damgård hashing for message authentication, *Advances in Cryptology – Asiacrypt 2007*, LNCS 4833, Springer-Verlag, 2007, pp. 216-231.

Department of Mathematics, Box 354350, University of Washington, Seattle, WA 98195 U.S.A.
E-mail address: koblitz@uw.edu

Department of Combinatorics & Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1 Canada
E-mail address: ajmeneze@uwaterloo.ca