“Practical IP” $\subseteq$ MA

Gilles BRASSARD †

Université de Montréal
Département IRO
C.P. 6128, Succ. "A"
Montréal (Québec)
CANADA H3C 3J7

Ivan Bjerre DAMGAARD ‡

Aarhus University, Mathematical Institute
Ny Munkegade
DK 8000 Aarhus C
DENMARK

Interactive protocols [GMR] and Arthur-Merlin games [B] have attracted considerable interest since their introduction a few years ago. These notions make it (probably) possible to extend the concept of what is “efficiently” provable to include, for instance, graph *non*-isomorphism [GMW]. In this short note, we assume that the reader is familiar with interactive protocols, Arthur-Merlin games, and the notion of zero-knowledge [GMR].

In the previous paragraph, we put quotes around “efficiently” because it is only the Verifier that is required to be efficient (i.e.: polynomial time). On the other hand, both interactive protocols and Arthur-Merlin games allow the Prover (or “Merlin”) to be infinitely powerful. In fact, not only is the Prover *allowed* to be powerful but she is actually *required* to be so in many of the most interesting theorems concerning these notions [B, GS, F, etc.]. For instance, in the graph non-isomorphism protocol, the Prover must be capable of deciding graph isomorphism.

An important pair of results state that $\mathrm{MA} \subseteq \mathrm{AM} = \mathrm{IP}[k]$ [B, GS], but again this requires the Prover to have considerable computing power *even if the original* MA *protocol is feasible*! From a practical point of view, this is silly in the sense that a polynomial-time Prover can run an MA protocol if only given a polynomial piece of advice, whereas it is not at all clear that she could run the corresponding AM protocol without additional power and/or information. (This is because the Prover must be able to satisfy exponentially many challenges in an AM setting.)

† Supported in part by Canada NSERC grant A4107

‡ Supported by the Danish Natural Science Research Council

S. Goldwasser (Ed.): Advances in Cryptology - CRYPTO ’88, LNCS 403, pp. 580-582, 1990.

For this reason, it is our opinion that **MA** is the natural extension of **NP** to randomness. This opinion is not new: it was already voiced in [BC]. However, here we claim that this is not merely an opinion but actually a theorem, *albeit* a rather trivial one. To achieve this goal, of course, we must be more precise on what we mean by "Practical **IP**": it is the class of languages that can be handled when both the Prover and the Verifier are restricted to being polynomial time.

This definition raises an important issue: if Prover and Verifier have similar computing abilities (and algorithmic knowledge), how did the Prover manage to obtain a hard enough proof to be of interest to the Verifier? (It is obviously *un*interesting if the Verifier can figure out the proof by himself.) One possible answer is that the Prover was lucky enough or that she worked hard enough to find it (this would presumably be the case for an eventual proof of FLT). A much more interesting answer, at least in cryptographic settings, is that the Prover obtained the statement of her claim *together with its proof*, as a result of running a probabilistic polynomial-time process (starting from some randomly chosen trap-door information). For instance, if the Prover wants a statement of the general form "the integer n is the product of exactly two distinct primes", she can simply choose the primes at random and multiply them. She then knows the factors of the result even though she is not better than the Verifier at factoring large integers. Read [AABFH] for a very nice theory on the efficient generation of solved hard instances of problems in **NP**.

Whatever is the origin of the information that allows the polynomial-time Prover to run her share of the interactive protocol, that information is necessarily polynomial in length. It is therefore reasonable to assert that "Practical **IP**" is *included* in "Polynomial-time **IP** with polynomial advice for the Prover" (**PIP**/Poly), where of course "polynomial-time" restricts both the Prover and the Verifier. (We are not willing to claim that "Practical **IP**" = **PIP**/Poly because in our view the really practical case for cryptography is when the advice comes from trap-door information rather than hard labour or luck.) Therefore, in order to prove the assertion given in the title of this paper, it suffices to prove that **PIP**/Poly $\subseteq$ **MA** (in fact, these classes are equal, but the reverse inclusion is irrelevant for our purpose).

Consider a language L in **PIP**/Poly, some x in L, and the polynomial-length advice a that the (polynomial-time) Prover could use through an **IP** to convince the Verifier that x is in L. The fact that L belongs to **MA** is obvious: given only x, an all-powerful Prover (Merlin) can figure out this advice a and simply give it to the Verifier (Arthur). The Verifier can then (in polynomial time) simulate the polynomial-time Prover and her interaction with him. This complete the proof that "Practical **IP**" $\subseteq$ **MA**. An open question is whether the inclusion is strict: in particular, is it possible in general to generate solved hard instances for every hard languages in **MA**? The reader is referred once more to [AABFH] for preliminary results concerning **NP**.

An interesting situation occurs if one is interested in zero-knowledge protocols [GMR]. It is shown in [BCC] (under cryptographic assumptions) that MA protocols can be carried out in zero-knowledge *by a polynomial-time Prover* provided she is given the corresponding piece of advice. This is in sharp contrast with the result of [GMW] in which an MA protocol must first be transformed into an AM protocol before it can be carried out in zero-knowledge, hence even a practical MA protocol requires a powerful Prover to be carried out in zero-knowledge if the technique of [GMW] is used. (This situation was already pointed out in [BCC].) In conclusion, [BCC] allows us to claim that

"Practical IP" = "Practical zero-knowledge",

which is the "practical" version of "everything provable is provable in zero-knowledge" [IY, BGGHKMR].

References

[AABFH]
Abadi, M., E. Allender, A. Broder, J. Feigenbaum and L. Hemachandra, "On generating hard instances of problems in NP", these *CRYPTO 88 Proceedings.*

[Ba] Babai, L., "Trading group theory for randomness", *Proceedings of the 17th Annual ACM Symposium on the Theory of Computing*, 1985, pp. 421-429.

[BGGHKMR]
Ben-Or, M., O. Goldreich, S. Goldwasser, J. Hastad, J. Kilian, S. Micali and P. Rogaway, "Everything provable is provable in zero-knowledge", these *CRYPTO 88 Proceedings.*

[BCC] Brassard, G., D. Chaum and C. Crépeau, "Minimum disclosure proofs of knowledge", *Journal of Computer and System Sciences*, to appear, 1988.

[BC] Brassard, G. and C. Crépeau, "Non-transitive transfer of confidence: a *perfect* zero-knowledge interactive protocol for SAT and beyond", *Proceedings of the 27th Annual IEEE Symposium on the Foundations of Computer Science*, October 1986, pp. 188-195.

[F] Fortnow, L., "The complexity of perfect zero-knowledge", *Proceedings of the 19th Annual ACM Symposium on the Theory of Computing*, May 1987, pp. 204-209.

[GMW] Goldreich, O., S. Micali and A. Wigderson, "Proofs that yield nothing but their validity and a methodology of cryptographic protocol design", *Proceedings of the 27th Annual IEEE Symposium on the Foundations of Computer Science*, October 1986, pp. 174-187.

[GMR] Goldwasser, S., S. Micali and C. Rackoff, "The knowledge complexity of interactive proof-systems", *Proceedings of the 17th Annual ACM Symposium on the Theory of Computing*, 1985, pp. 291-304.

[GS] Goldwasser, S. and M. Sipser, "Arthur-Merlin games versus interactive proof systems", *Proceedings of the 18th Annual ACM Symposium on the Theory of Computing*, May 1986, pp. 59-68.

[IY] Impagliazzo, R. and M. Yung, "Direct minimum-knowledge computations", *Proceedings of CRYPTO 87*, August 1987, pp. 40-51.