On using probabilistic Turing machines to model participants in cryptographic protocols[*]

Lee Klingler
Florida Atlantic University

Rainer Steinwandt
Florida Atlantic University

Dominique Unruh
University of Tartu, Estonia

August 14, 2013

Abstract

To formalize participants in cryptographic protocols, it is common to use probabilistic Turing machines. We point out subtleties in common definitions of probabilistic Turing machines, which imply that the common cryptographic operation of uniform random sampling in a finite set $\{1, \ldots, s\} \subseteq \mathbb{Z}$ is in general not possible within this model. From a technical point of view, this invalidates in particular a standard proof of the perfect zero knowledge property of the popular graph isomorphism proof system. The observed limitation appears to be relevant for other cryptographic protocol analyses as well, and we suggest one possible tweak of the definition of a probabilistic Turing machine.

Keywords: Provable security, Turing machine, $\mathbb{Z}$-algebra.

1 Probabilistic Turing machines revisited

Cryptographic protocols and adversary models are almost always probabilistic. Therefore, the definition of probabilistic Turing machines constitutes the underpinning of most cryptographic security proofs. There are, however, subtleties in the definition of probabilistic Turing machines that are often ignored or argued to be irrelevant.

To illustrate the problem, consider the popular perfect zero-knowledge proof system for graph isomorphism [GMW86].

Example 1 (graph isomorphism proof system)**.** *Given two graphs G_0 and G_1 along with an isomorphism $\phi : G_0 \longrightarrow G_1$, the prover picks a uniformly random permutation ψ on the vertices of G_0 and computes $H := \psi(G_0)$. Then the prover sends H to the verifier (who knows G_0 and G_1 but not ψ). The verifier picks a bit $b \in \{0, 1\}$ uniformly at random and sends b to the prover. The prover responds with an isomorphism $\psi_b : G_b \longrightarrow H$ which can be efficiently computed from ϕ and ψ.*

[*] Appeared in Theoretical Computer Science [KSU13].

It is claimed [Gol01] that this proof system is perfect zero-knowledge in the sense of [Gol01]. That is, there is a polynomial-time simulator that (conditioned on not aborting) produces an execution transcript that has the same distribution as the interaction between prover and verifier. The simulator only has access to G_0 and G_1 but not to ϕ.

A technical point that is important to note here, is that both the honest prover and the simulator pick a uniformly random permutation on the set of vertices. In other words, given a number n (of vertices), they have to pick a permutation on n elements.

Can this can be done in polynomial time? The machine model typically used in cryptography are probabilistic Turing machines (PTM) following Gill [Gil74]. Such a *Gill-PTM* is defined like a deterministic Turing machine, except that the Gill-PTM additionally has access to an infinite sequence of uniformly and independently distributed bits. It is easy to see that, for a Gill-PTM running in time $\leq t$, the probability of any output is of the form $a/2^t$ for some $a \in \mathbb{N}_0$. In particular, such a probability cannot be exactly $1/6$. Since for $n = 3$, a uniformly chosen permutation is the identity with probability $1/(3!)$ $=1/6$, it follows that in time t (for any integer t), it is not possible to pick a uniform permutation on three elements. In particular, the prover and the simulator for the graph isomorphism protocol cannot be implemented in polynomial-time.[1]

Remark 1. *Note that it is easy to approximate the prover and the simulator: We can come arbitrarily close to the right distribution. But for the purposes of* perfect *zero-knowledge, approximating the prover and simulator is of no use. We need exact equality of distributions.*

Thus the proof from [Gol01] that there are perfect zero-knowledge proofs for graph-isomorphism does, strictly speaking, fail. For the perfect zero-knowledge proof system for *Graph 3-Colorability* as described in [Gol01] we encounter a similar situation—here the prover has to select a random element in the symmetric group on three points, and the verifier has to select uniformly at random an edge of a given graph.

But let us step back and analyze where the problem comes from in the graph isomorphism example. We had to construct Turing machines that pick elements with probabilities that are not of the form $a/2^t$. And since a Gill-PTM inherently uses random bits, this is impossible. So why do Gill-PTMs have this (seemingly arbitrary) restriction to random *bits*? For example, given unbiased rolls of a fair six-sided die, we could pick a permutation on three elements. In [Gil74], Gill-PTMs were introduced with the purpose of defining the complexity class BQP. Since this class considers computations with bounded errors, assuming uniform *bits* is no restriction as other probabilities can be approximated sufficiently closely. Thus using random *bits* is a justified simplification in that setting, as well as in most situations in cryptography. When dealing with a setting where exact probabilities matter, this simplification is not justified. We have to go back

[1] That is, for any polynomial p, there is no Gill-PTM that, on input n, outputs a uniform permutation of n elements and that terminates with probability 1 in time $\leq p(n)$.

It is possible to chose a uniform permutation in *expected* polynomial time, though. For example, [GMW86] use expected polynomial-time simulators (for reasons unrelated to the sampling of permutations). Expected polynomial-time simulators lead to other difficulties though, see, e. g., [Gol07].

to the way in which PTMs were defined before [Gil74]. The original definition of PTMs seems to be Santos [San69]. A Santos-PTM is a Turing machine in which the state transition function (which determines what symbol to write on the tape, and in which direction to move) is described by a finite probabilistic automaton. That is, for each state and each symbol under the head, there is a probability distribution that specifies head movement, new symbol, and new state. When constructing a Santos-PTM, we can chose an arbitrary such distribution.[2] (But that distribution may not depend on the input of the Santos-PTM, of course.)

The question now arises: Is the proof system from [GMW86] and its simulator polynomial-time if we assume Santos-PTMs as our machine model? More precisely, we ask the following question:

> Is there a Santos-PTM that on input s outputs 1 with probability $1/s$ and that runs in time polynomial in s?

Remark 2. *It can be seen that a positive answer to this question is both sufficient and necessary for being able to pick permutations on n elements.*

First, we easily observe that such a Santos-PTM cannot exist if the probabilities are rational. Assume a Santos-PTM that runs in time $p(s)$. Let $a_1, \dots, a_n$ denote all probabilities occurring in the probability distribution of that Santos-PTM. Then the probability of output 1 on input s is the sum of the probabilities of all execution paths leading to output 1. Each such execution path has a probability which is a product of a_i. Furthermore, since the time is limited to $p(s)$, there are only finitely many different execution paths. Thus, the probability of output 1 is a polynomial expression in $a_1, \dots, a_n$ with integer coefficients. In other words, that probability lies in $\mathbb{Z}[a_1, \dots, a_n]$, the ring of all such polynomial expressions, called the $\mathbb{Z}$-algebra generated by $a_1, \dots, a_n$. If all a_i are rational, then any $r \in \mathbb{Z}[a_1, ..., a_n]$ can be expressed as a fraction with denominator which is a product of the denominators of the reduced a_i. In particular, if s is a prime not occurring in those denominators, then $1/s \notin \mathbb{Z}[a_1, \dots, a_n]$. Hence for any Santos-PTM with rational probabilities there always is an s such that the Santos-PTM does not output 1 with probability $1/s$. In fact, as there are only finitely many prime factors in the denominators of the reduced a_i, the Santos-PTM fails to output 1 with probability $1/s$ for almost all prime numbers s.

However, if we consider a Santos-PTM with irrational probabilities, the situation is not as simple. Perhaps there is a particular choice of values $a_1, \dots, a_n$ such that any $1/s \in \mathbb{Z}[a_1, \dots, a_n]$? In the next section, we will show that this is not the case and that thus even Santos-PTMs cannot be used for the proof system of graph isomorphism.

[2] As Bernstein and Vazirani [BV97, Section 3.1] observe, in a context of polynomial-time computations, one should however limit the probabilities to numbers that are computable to within 2^{-n} in time polynomial in n by a deterministic Turing machine.

2 Impossibility of random selection with arbitrary rational probabilities

Note that the fractions $1/s$, as s ranges over $\mathbb{N}$, generate $\mathbb{Q}$ as a $\mathbb{Z}$-algebra. That is, every polynomial expression in $1/s_1, \ldots, 1/s_n$ ($s_i \in \mathbb{N}$) with integer coefficients lies in $\mathbb{Q}$, and each element of $\mathbb{Q}$ has such a representation. Thus, to prove that there exists an $s \in \mathbb{N}$ with $1/s \notin \mathbb{Z}[a_1, \ldots, a_n]$, it is sufficient to establish the following result.[3]

Proposition 1. *Let $\{a_1, \ldots, a_n\} \subseteq \mathbb{R}$ be a finite set of real numbers. Then $\mathbb{Q} \nsubseteq \mathbb{Z}[a_1, \ldots, a_n]$.*

Proof. Denote by φ the evaluation map from the polynomial ring $\mathbb{Z}[x_1, \ldots, x_n]$ to the $\mathbb{Z}$-algebra $\mathbb{Z}[a_1, \ldots, a_n]$, that is, the map defined by $\varphi(p(x_1, \ldots, x_n)) = p(a_1, \ldots, a_n)$. Clearly φ is a ring homomorphism and maps $\mathbb{Z}[x_1, \ldots, x_n]$ onto $\mathbb{Z}[a_1, \ldots, a_n]$, so by a standard isomorphism theorem $\mathbb{Z}[a_1, \ldots, a_n]$ is isomorphic to the factor ring $\mathbb{Z}[x_1, \ldots, x_n]/\ker(\varphi)$, where $\ker(\varphi)$ denotes the kernel of φ, which is the subset of those $p \in \mathbb{Z}[x_1, \ldots, x_n]$ such that $\varphi(p) = 0$. Let M be a maximal ideal of the ring $\mathbb{Z}[x_1, \ldots, x_n]$ containing $\ker(\varphi)$.[4] By a standard homomorphism theorem, there is a ring homomorphism π from the factor ring $\mathbb{Z}[x_1, \ldots, x_n]/\ker(\varphi)$ onto the factor ring $K := \mathbb{Z}[x_1, \ldots, x_n]/M$, defined by mapping the coset $p + \ker(\varphi)$ to the coset $p + M$.

We claim that $\mathbb{Z} \cap M \neq \{0\}$. The factor ring K is generated as a $\mathbb{Z}$-algebra by the finitely many cosets $x_1 + M, \ldots, x_n + M$, and K is a field (because M is a maximal ideal). If $\mathbb{Z} \cap M = \{0\}$, then the cosets of the integers would all be distinct in K, so that, since K is a field, it would contain a subfield isomorphic to $\mathbb{Q}$. Since K is finitely generated as a $\mathbb{Z}$-algebra, it would then be finitely generated as a $\mathbb{Q}$-algebra, so [AM69, Proposition 7.9] would force K to be a finite dimensional vector space over $\mathbb{Q}$. Then from [AM69, Proposition 7.8], it would follow that the rational numbers are finitely generated as $\mathbb{Z}$-algebra, which is clearly not the case, as noted above. This proves the claim that $\mathbb{Z} \cap M \neq \{0\}$.

Therefore, there is a nonzero integer $m \in M$, which means the coset $m + M = 0 + M$, the zero element of the field K. Then the ring homomorphism π maps the coset $m + \ker(\varphi)$ (the sum of m terms $\pi(1 + \ker(\varphi))$) to $m + M$ (the sum of m terms $1 + M$), by definition of a ring homomorphism. That is, π maps $m + \ker(\varphi)$ to the zero element of K, so that $m + \ker(\varphi)$ can have no reciprocal in $\mathbb{Z}[x_1, \ldots, x_n]/\ker(\varphi)$, because the ring homomorphism π must preserve products and the multiplicative identity. Identifying integers with their cosets in $\mathbb{Z}[x_1, \ldots, x_n]/\ker(\varphi)$, it follows that m has no reciprocal in $\mathbb{Z}[x_1, \ldots, x_n]/\ker(\varphi)$, so that $\mathbb{Q}$ cannot be contained in $\mathbb{Z}[x_1, \ldots, x_n]/\ker(\varphi)$. Since $\mathbb{Z}[a_1, \ldots, a_n]$ is isomorphic to $\mathbb{Z}[x_1, \ldots, x_n]/\ker(\varphi)$, the proposition follows. □

[3] The proof is not difficult, but we did not find it in the literature.

[4] A maximal ideal is a proper ideal which is not contained in any other proper ideal. The existence of a maximal ideal containing $\ker(\varphi)$ follows from Zorn's Lemma [AM69, Theorem 1.3].

3 Conclusion

We have shown that the modeling of probabilistic Turing machines by both Gill [Gil74] and Santos [San69] do not precisely capture the way Turing machines are used in cryptographic analyses dealing with perfect security. To address this issue, we propose to use a model of probabilistic Turing machines that allows one to pick random elements from finite sets whose size is not a power of 2:

Definition 1. *A* PTM with sampling oracle *is a deterministic Turing machine endowed with a probabilistic oracle $\mathcal{O}$ such that, upon invocation $\mathcal{O}(r, s)$ with $r/s \in [0, 1]$, the oracle returns* 1 *with probability r/s; otherwise* 0 *is returned. Here r and s are assumed to be encoded in binary, so that invoking the oracle takes time* $\mathrm{O}(\log(r) + \log(s))$.

It is easy to see that with such an oracle, we can pick uniformly random elements from $\{1, \dots, s\}$ in time $\mathrm{polylog}(s)$, as well as from the set of permutations on s elements in time $\mathrm{poly}(s)$. Thus PTMs with sampling oracle are suitable, e. g., for the zero-knowledge proof for graph isomorphism.

Acknowledgments Dominique Unruh was funded by institutional research grant IUT2-1 from the Estonian Research Council, and by European Regional Development Fund and the Estonian ICT program 2011-2015 (3.2.1202.12-0001), by the European Social Fund's Doctoral Studies and Internationalisation Programme DoRa, by the European Regional Development Fund through the Estonian Center of Excellence in Computer Science, EXCS.

References

[AM69] M. F. Atiyah and I. G. MacDonald. *Introduction to Commutative Algebra.* Addison-Wesley Series in Mathematics. Addison-Wesley, 1969.

[BV97] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM J. Computing*, 26(5):1411–1473, 1997. Online available at `http://dx.doi.org/10.1137/S0097539796300921`.

[Gil74] John T. Gill, III. Computational complexity of probabilistic Turing machines. In *Proceedings of the sixth annual ACM symposium on Theory of computing*, STOC '74, pages 91–95, New York, NY, USA, 1974. ACM.

[GMW86] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. In *27th Annual Symposium on Foundations of Computer Science, Proceedings of FOCS 1986*, pages 174–187. IEEE Computer Society, 1986. Online available at `http://www.wisdom.weizmann.ac.il/~oded/X/gmw1c.pdf`.

[Gol01] Oded Goldreich. *Foundations of Cryptography: Basic Tools*, volume 1. Cambridge University Press, Cambridge, UK, 2001.

[Gol07] Oded Goldreich. On Expected Probabilistic Polynomial-Time Adversaries: A Suggestion for Restricted Definitions and Their Benefits. In Salil Vadhan, editor, *Theory of Cryptography, Proceedings of TCC 2007*, Lecture Notes in Computer Science, pages 174–193. Springer-Verlag, 2007. Online available at `http://eprint.iacr.org/2006/277.ps`.

[KSU13] Lee Klingler, Rainer Steinwandt, and Dominique Unruh. On using probabilistic turing machines to model participants in cryptographic protocols. *Theoretical Computer Science*, 501:49–51, 2013.

[San69] Eugene S. Santos. Probabilistic Turing Machines and Computability. *Proceedings of the American Mathematical Society*, 22(3):pp. 704–710, 1969.