The Foundations of Cryptography's "LaTeX-ized" ERRATA

Oded Goldreich & others (the actual errata)
Andrea Barontini* (present document editing)

October 30, 2021

1 Foreword

Errata are an often neglected part of books life-cycle, even if they are what keeps contents alive and valuable through years. Luckily the insightful Foundations of Cryptography by Oded Goldreich doesn't miss the point, however while reading it as autodidact I have been desiring a bit more formatting than what is possible by means of a plain web page interleaved with some un-compiled LaTeX code. So I have copied & pasted **List of Corrections** from the volumes companion pages (https://www.wisdom.weizmann.ac.il/~oded/foc-vol1.html#err and https://www.wisdom.weizmann.ac.il/~oded/foc-vol2.html#err, retrieved on Oct 26th, 2021) and slightly edited them -without modifying contents- to take advantage of some reader-oriented commodities:

- formulas are, of course, rendered (perhaps the main reading improvement);
- italic and bold texts, explicit newlines, hyperlinks have been maintained;
- all hyperlinks have been updated to https;
- hyperlinks target URLs show up in new footnotes, to not loose those infos when printing;
- lists are numbered (just to make easier here to reference specific items);
- custom command `\poly` used in items 18b and 33 has been defined as `\mathrm{poly}`;
- custom command `\xor` used in item 19 has been defined as alias to `\oplus` to get the same symbol as in the original formula;
- left opening round brackets `(` corrected with braces `{` in fractions denominators of item 23;
- to honor book style, custom command `\pr` in item 27 has been defined as `\mathsf{Pr}` instead of replacing it with standard `\Pr`;

*andrea.barontini@bybaro.it

- added missing $ delimiting formulas in item 31;
- removed an extra $ in item 37 causing wrong parsing of following text;
- just some minor and indubitable typos corrected: *permuation* (item 5), *otherwiose* (item 13), *concatanation* (item 22b), *computible* and *compurted* (item 33), *completenss* (item 37), *descryption* (item 42).

2 List of Corrections

2.1 Volume I - Basic Tools

The following errors were corrected in Volume 2[1]; for further details see Appendix C (of Volume 2[1]) as well as a draft of the said appendix[2], Feb. 2003.

1. Unfortunately, we have to withdraw two claims regarding *strong* witness indistinguishable proofs as defined in Definition 4.6.2. Specifically, in general, *strong* witness indistinguishability is not closed under parallel composition (and so Lemma 4.6.7 is wrong). Consequently, in contrary to what is stated in Theorem 4.6.8, we do not know whether there exist constant-round public-coin proofs with negligible error that are *strong* witness indistinguishable for languages out of BPP. We stress that the flaws pointed out here only refer to *strong* witness indistinguishability and not to (regular) witness indistinguishability. That is, as stated in Lemma 4.6.6, (regular) witness indistinguishability is closed under parallel composition and thus the part of Theorem 4.6.8 that refers to regular witness indistinguishability is valid (i.e., providing constant-round public-coin proofs with negligible error that are witness indistinguishable for NP).

2. Remark 4.10.6 as well as Theorems 4.10.10, 4.10.14 and 4.10.16, seem to require trapdoor permutations in which the permutation's domain coincides with the set of all strings of certain length. This special case of Definition 2.4.5 can be implemented by modifying the RSA or the Factoring Trapdoors (cf. Canetti et al (in STOC'96) and [5]). (For further detail on Remark 4.10.6, the reader is referred to [23].)

   Added comment (Nov. 2008): Unfortunately, the correction that appeared in Appendix C of Volume 2[1] is also incorrect. For details see note[3] (or better its revision[4]).

   For a discussion of *enhanced trapdoor permutations* see my paper with Ron Rothblum titled Enhancements of Trapdoor Permutations[5].

3. Not exactly an error, yet a recent result by Boaz Barak (see FOCS'01) calls for re-evaluation of the significance of all negative results regarding black-box zero-knowledge (cf. Definition 4.5.10). In particular, relying on standard intractability assumptions, Barak presents round-efficient public-coin zero-knowledge arguments for NP (using non-black-box simulators),

[1] https://www.wisdom.weizmann.ac.il/~oded/foc-vol2.html
[2] https://www.wisdom.weizmann.ac.il/~oded/PSBookFrag/errI.ps
[3] https://www.wisdom.weizmann.ac.il/~oded/PSBookFrag/nizk-tdp.ps
[4] https://www.wisdom.weizmann.ac.il/~oded/COL/nizk-tdp.pdf
[5] https://www.wisdom.weizmann.ac.il/~oded/p_tdp.html

whereas only BPP can have such black-box zero-knowledge arguments (cf. comment following Theorem 4.5.11). Interestingly, Barak's simulator works in strict (rather than expected) probabilistic polynomial-time, addressing an open problem mentioned in Section 4.12.3.

4. In Definition 4.10.15 (adaptive NIZK), the adaptive zero-knowledge condition (page 310) should relate *only* to input-selecting functions Ξ that are implementable by a family of polynomial-size circuits.

5. Const. 4.10.7 and Prop. 4.10.9 (pp. 303-304): The current description in terms of two mappings π_1, π_2 is confusing and even inaccurate. Instead one should identify the row and columns of H with $[n]$ and use one permutation/isomorphism π of G to H. Alternatively, one may compose this π with the ψ_i's (which results in π_i's as in the text). [Luis von Ahn]

6. Definition 4.9.3 is carelessly written. One should partition the commit phase into two sub-phases, with the second sub-phase being a proof-of-knowledge referring to the view of the first sub-phase, which in turn should constitute a commitment scheme by itself. Alternatively, a Non-Oblivious Commitment may be defined as the sequential composition of an ordinary commitment scheme with a corresponding proof-of-knowledge such that the composed protocol is also a commitment scheme. [Yehuda Lindell]

7. The proof of Theorem 4.9.4 is wrong, since it relies on the existence of constant-round strong witness indistinguishability proofs of knowledge for NP (i.e., Theorem 4.6.8). Fortunately, an alternative proof exists (see Apdx C.3.3 in Volume 2^1).

8. Addition to Section 4.12.2: In continuation to Sections 4.7 and 4.9.2, we mention that the round-efficient argument system of [77] is actually an "argument of knowledge" (with negligible error).

9. Adding self-credits to Section 4.12.1: the notions of *strong* witness indistinguishability (Section 4.6) and *strong* proofs of knowledge (Section 4.7.6), and the Hidden Bit Model (Section 4.10.2) are due to (early versions of) this work.

The foregoing errors were corrected in Volume 2^1. More recent corrections and additions follow

10. The specific expression given for Chernoff Bound (in Sec 1.2.2, page 11) may be wrong. The exponent should be either $-(\epsilon^2/(4p(1-p))) \cdot n$ (i.e., a factor of two smaller than stated) or just $-2\epsilon^2 \cdot n$. (Both alternatives are correct, but one usually sees the latter.) [Kivanc Mihcak]

11. Typo on page 35, 2nd paragraph: It should be $\nu(n) > 1/p(n)$. [Eldad Zinger]

12. There is a typo on page 46 (just before equation 2.7): N0 should be N'.

13. Formally speaking, Prop. 2.4.1 (as stated) is trivial. The statement asserts that there exists a poly-time computable f such that f is a OWF iff OWF exist. But, as pointed out by Claus Diem, if OWF exist then one may use f as any of them, and otherwise one may use any poly-time computable f.

Of course, what I meant was that one can specify a poly-time computable f without knowing if OWF exist or not, but this is not captured by the formal statement. Of course, one way out is to describe f first, and then make the claim about this f; but, for obvious reasons, I did not want to do that.

14. In continuation to Sec 2.4.2 and 2.4.4, see clarifications regarding one-way permutations[6], 2005.

15. The lower bound (on the size of S_n) in Claim 2.5.2.1 can be improved to $\epsilon(n) \cdot 2^n$. See details HERE[7]. [Noam Livne]

16. Section 2.5.3 does not provide the best analysis of the security of hard-core functions. A better analysis is provided HERE[8].

17. At the end of Section 2.7.3, the question should refer to returning a constant fraction of the bits of x (rather than the first half of x).

18. The presentation in Section 2.6.2 contains a (fix-able) error. The graph $G_{f,n}$ defined on page 82 is a directed graph, and the construction of the function F (Construction 2.6.3) refers to directed walks on it. But Lemma 2.6.5 refers to undirected graphs and hence cannot be applied to it. [Qifu Hu]

    (a) Nevertheless, the weaker bound provided in the proof of Lemma 2.6.5 holds also in this case. Specifically, consider the stochastic process captured by the matrix M_f such that $M_f = MR$, where M is the normalized adjacency matrix of the graph G_n and R is the permutation matrix that corresponds to the mapping $x \mapsto f(x)$. Now we want to bound $\|(PMR)^t z\|$ and we do it by bounding $\|PMRz\|$, which reduces to bounding $\|PMz'\|$ for $z' = Rz$ (while noting that Rz and z have the same norm). Hence, we obtain a bound of $1 - (1 - 0.5\mu)^{t/2}$, rather than $1 - (1 - 0.5\mu)^t$, and this is the bound we should use for $\beta(n)$ (with $t = k(n)/\ell$) when we invoke Lemma 2.6.6. This means that we establish the corresponding bound of with $\beta(n) = 1 - (1 - 0.5\alpha(n))^{k(n)/2\ell}$ also in Prop. 2.6.4 (i.e., we lose a factor of two in the exponent).

    (b) Note, however, that the above analysis presumes that $\mu \geq 2\rho^2$, where ρ is an upper bound on the eigenvalue ratio. This is fine when applying Lemma 2.6.6 with a constant value of α, but for smaller values we have to use an expander of degree $d = \text{poly}(1/\alpha)$ and a polynomially related expansion bound (where such an expander can be obtained by talking a walk of length $\log d$ on a constant-degree expander). This in turn means that $\ell = O(\log(1/\alpha))$ and that we can only use $k(n) = O(n/\ell)$ in Prop 2.6.4 (in order to maintain $n + k(n) \log d = O(n)$). This only means that in the first applications of Prop 2.6.4 (within the proof of Thm 2.6.2), we only increase the hardness parameter by a factor of $n/\log n$ (rather than by n). That

[6] https://www.wisdom.weizmann.ac.il/~oded/owp.html
[7] https://www.wisdom.weizmann.ac.il/~oded/PSBookFrag/add3.txt
[8] https://www.wisdom.weizmann.ac.il/~oded/PSBookFrag/add2.txt

is, starting with an $n^{-\delta}$-OWP, for $i \in [\delta]$, after i iterations we derive an $(n^{-(\delta-i)}/(\log n)^i)$-OWP, and in the last (i.e., $\delta+1$-st) iteration we obtain a strong one-way function (since $1-(1-(1/\text{poly}\log n))^{n/\log n}$ is negligible).

I'm currently unsure if the original claim (i.e., with Lemma 2.6.5 applied to the directed graph $G_{f,n}$) is true, but it does not matter much.

19. There is a typo in the guideline given for Exercise 9 of Chapter 2: The suggestion should be $f(y,z) = (g(y), y \oplus z)$.

20. There is a typo in the guideline given for Exercise 17 of Chapter 2: The statistical distance is stated at the end should by $O(2^{-\ell} + 2^{2\ell+2\log n} \cdot \epsilon)$.

21. Typo in page 99 line 3 (Exer 28 in Chap 2): the reference should be to [204] not [203].

22. Clarification for Exer 28 of Chap 2:

    (a) The notation $\ell(n)$ is abused; in Item 2 it denotes the codeword length, which is exponential in the length of the description of a position in the codeword (as used in Item 3).

    (b) In Item 3, use a concatenation code in which symbols of the Reed-Solomon code are encoded via the hadamard code.

23. Prop. 3.3.8 (on page 124) can be extended to all pseudorandom generators G of arbitrary stretch $ell(n)$, yielding a strong OWF if $\ell(n)-n = \omega(\log n))$ and a weak OWF otherwise. This follows by observing that for any subset S of the image of G (or rather the image of n-bit long seeds under G), it holds that $prob[G(U_n) \in S] \geq \frac{|S|}{2^n}$ whereas $prob[U_{ell(n)} \in S] = \frac{|S|}{2^{ell(n)}} \leq \frac{|S|}{2^{n+1}}$. [Guang Yang]

24. The sampling procedure referred to at the end of Sec 3.4.2 is not as obvious; the $O(n^3)$ bound relies on the fact that the expected number of candidates to be tried is $O(n)$. [Qifu Hu]

25. Sec. 3.5.1.2 (on page 138), right after Prop 3.5.3, contains a typo: k should be set to $O(p(n))$, not to n. [Qifu Hu]

26. Typo on page 142 (last sentence of 2nd paragraph): It should be m not i. [Qifu Hu]

27. Typo on page 143 (1st paragraph of the proof of Prop 3.5.9): The 3rd sentence should have been "Thus, for every y and α, it holds that $\mathsf{Pr}[|F^{-1}(y,\alpha,H_n^{m(n)-l(n)})| > 2^{l(n)+1}] < 2^{-l(n)}$." and the rest of this paragraph should have referred to $B_y = \{(\alpha,h) : |F^{-1}(y,\alpha,h)| > 2^{l(n)+1}\}$. [Qifu Hu]

28. Sec. 3.5.3 (on page 147) contains an error. The number of copies should be n^3, not n^2. In all expressions in Sec. 3.5.3, n^2 should be replaced by n^3, and n^3 by n^4. [Yu Yu]

29. Unfortunately, Theorem 3.5.12 (i.e., OWF implies PRGs) still does not have an easy proof (i.e., one suitable for a textbook). But proofs are getting easier and better. Currently, the most reader-friendly proof can be found in Efficiency Improvements in Constructing Pseudorandom Generators from One-way Functions (by Iftach Haitner, Omer Reingold, and Salil Vadhan)[9].

30. Typo on page 155 at the beginning of the proof of Claim 3.6.6.1. The queries in the simple case should be the **reverses** of the first t strings in lex-order; that is, for $t = 2^{k+1}$, we use strings of the form $x'0^{n-k-1}$, where $x \in \{0,1\}^{k+1}$. [Qifu Hu]

31. Typo on page 156 line 12: The text should read "In fact, $g(p_i) = g(p_j) = v_i = v_j$ also in case $p_i \neq p_j$" (i.e., inequality rather than equality). [Qifu Hu]

32. Typo on Page 163, in the displaced definition of f: In the otherwise-case, one should replace $G_{\sigma_{k+1}}$ by $P_{\sigma_{k+1}}$, where $P_0(s)$ (resp., $P_1(s)$) is defined as the first n bits of s (resp., the next n bits of s), just as in the proof of Thm 3.6.6. [Qifu Hu]

33. The definitions in Sec 3.6.1 fail to require that the function ℓ is upper bounded by a polynomial. Furthermore, they also fair to require that ℓ is efficiently computable (i.e., $\ell(n)$ can be computed in poly(n)-time when given n). [Dima Kogan and Inbal Livni]

    Both conditions are not used in the text, but I agree that it is a good idea to impose them both. I tend to insert these conditions into Def 3.6.1 and the preamble of Def 3.6.4. Alternatively, it may be inserted into Def 3.6.3 and the first item of Def 3.6.4.

34. There is a typo in Exercise 28 of Chapter 3: The threshold value should be $2^{e}ll \cdot ell$ (rather than ℓ^{ℓ} as written).

35. Regarding Footnote 7 in Sec 4.3.1.1 (page 201), see a recent study of Super-Perfect Zero-Knowledge Proofs (2014)[10].

36. In continuation to the definition of Proofs Of Knowledge (Sec 4.7.1), see discussion of [10]Probabilistic versus Deterministic Provers in the Definition of Proofs Of Knowledge[11], 2006.

37. Thm 4.4.11 fails to clarify that this result holds for any NP-witness relation, and not merely for any NP-set [Roei Tell]. Likewise, in Def 4.3.10, it would have been better to define (efficient) zero-knowledge interactive proofs for any relation R rather than refer to the set of strings $P_L(x)$ that satisfy completeness. In such case, the completeness and zero-knowledge conditions should hold for any $(x,y) \in R$ (rather than for every $x \in L$ and $y \in P_L(x)$), for any R such that L contains all x's for which there exists a y such that $(x,y) \in R$.

[9]https://www.wisdom.weizmann.ac.il/~oded/X/hill-revisited.pdf
[10]https://www.wisdom.weizmann.ac.il/~oded/p_spzk.html
[11]https://www.wisdom.weizmann.ac.il/~oded/p_pok-note.html

38. Remark 4.10.6 as well as Theorems 4.10.10, 4.10.14 and 4.10.16, seem to require trapdoor permutations in which the permutation's domain coincides with the set of all strings of certain length. This special case of Definition 2.4.5 can be implemented by modifying the RSA or the Factoring Trapdoors (cf. Canetti et al (in STOC'96) and [5]). As stated above, the correction that appeared in Appendix C of Volume 2[1] is also incorrect. For details see note[4].

39. In Construction 4.10.12, it seems better to use the symbol u' (in the definition of L_2) instead of w'. The point is to streamline better with the proof of Proposition 4.10.13.

40. A recent work of Nguyen, Ong, and Vadhan (see 47th FOCS, 2006) provides *statistical zero-knowledge arguments for NP from any one-way function*. This resolves a central open problem in the area (also mentioned in Sec. 4.12.3). See also a follow-up work by Haitner and Reingold (see 39th STOC, 2007) providing *statistical hiding commitment schemes*.

41. Exercise 4 of Chapter 4 (p. 324) is wrong. The guideline establishes that this "shared randomness interactive proof" model is at most as powerful as AM, and in fact it coincides with AM, where a set S is in AM if there exists a set S' in NP and a polynomial p such that for every $x \in S$ it holds that $Prob_{r:|r|=p(|x|)}[(x,r) \in S'] \geq 2/3$, and for every $x \in S$ it holds that $Prob_{r:|r|=p(|x|)}[(x,r) \notin S'] \geq 2/3$. [Ben Diamond]

2.2 Volume II - Basic Applications

42. On page 436, Item 1, the description fits the private-key case. In the public-key case, A_1 should be given input (e,z), rather than $(1^n,z)$. The same holds for Item 1 on page 443. [Rupeng Yang]

43. On page 443, Item 1: See page 436, Item 1.

44. On page 475: All mentions of the *authenticated key-exchange* should be *unauthenticated key-exchange*, and vice verse. [Haiyang Xue]

45. On page 481, Footnote 58, the Decisional Diffie Hellman problem is wrongly stated. One should require instead that $P = 2P' + 1$, with P' being a prime, and that g is a generator of the set of quadratic residues mod P. [Luca Trevisan; for further details, see Luca's email[12].]

46. Addendum to the guideline of Exer 5.2 (on page 482): To show that the two distributions are statistically far apart, consider the event S such that $(r,s) \in S$ if and only if there exists an n-bit string K such that $D_K(s) = r$.

47. On top of page 622, the simulator constructed to demonstrate the point should only output the view s, rather than the view coupled with the output $(s, F(s))$. [Carmit Hazay]

48. On *constructing 1-out-of-k OT* (Const. 7.3.5 and Prop 7.3.6). The current description is fine if $k = 2$, which actually suffices via additional reductions. For $k > 2$ one should either modify Construction 7.3.5 or make

[12] https://www.wisdom.weizmann.ac.il/~oded/PSBookFrag/luca-ddh.txt

stringer assumptions about the trapdoor permutation used. The simplest modification is to let the sender select k indices of permutations (rather than a single one), and have the parties use the ith index for x_i and y_i. [Ron Rothblum]

49. On top of page 698, item 1 (in the discussion of the (first) malicious model) refers to a convention made in Sec. 7.2.3 (which can be found in Footnote 17 on page 628). As pointed out by Yuval Ishai, in the multi-party context, it seems that the special abort-symbol convention (including a modified functionality) has to be used. In retrospect, we prefer this convention also in the two-party case.

50. Regarding the issue of **fairness** (cf. Sec. 7.2.3 and 7.7.1.1): See an important revisiting of this issue in Complete Fairness in Secure Two-Party Computation[13] by D. Gordon, C. Hazay, J. Katz, and Y. Lindell (STOC'08, pp. 413-422).

51. Errata regarding the paragraph following Thm 7.5.14 (on p. 708) and the 2nd paragraph on the 1st item on page 709: This assertion is correct only if each output bit in the circuit is a non-linear function of each of the input bits, which means that a multiplication gate appears on each input-output path. This can be enforced by augmenting the circuit with gates that multiply each input bit by itself (or multiply each bit output by itself before outputting it). [Peter Scholl]

52. Regarding Section 7.6: See a full proof of the BGW Protocol for perfectly-secure multiparty computation[14], provided by Gilad Asharov and Yehuda Lindell, 2011.

53. Regarding *NIZK and enhanced trapdoor permutations* (see Appendices C.1 and C.4.1): Jonathan Katz has pointed out that although the notion of enhanced trapdoor permutations (Def. C.1.1) suffices for constructing Oblivious Transfer (see Sec. 7.3.2), it does not seem to suffice for constructing a NIZK (as outlined in Remark 4.10.6 and patched in Apdx C.4.1). A seemingly stronger notion that does suffice (for constructing a NIZK) may be called a *doubly enhanced trapdoor permutation* and is defined as an enhanced trapdoor permutation for which given an n-bit long description of a permutation it is feasible to generate a random pair (x, r) such that x is the preimage under the permutation of the domain element generated by the domain-sampling algorithm on coins r. For further details, see note [Nov. 2008 (rev'ed Oct. 2009)][3].

    More about *enhanced trapdoor permutations*: See Ron Rothblum's Taxonomy of Enhanced Trapdoor Permutations[15], 2010.

    Actually, it may be best to start with a later paper by Ron any myself, titled Enhancements of Trapdoor Permutations[5].

[13] https://www.wisdom.weizmann.ac.il/~oded/X/fairness.pdf
[14] https://www.wisdom.weizmann.ac.il/~oded/MC/067.html
[15] https://eccc.weizmann.ac.il/report/2010/145/
(original http://eccc-preview.hpi-web.de/report/2010/145/ not working anymore)