

A Pragmatic Introduction to Secure Multi-Party Computation: Errata

David Evans

University of Virginia
evans@virginia.edu

Vladimir Kolesnikov

Georgia Institute of Technology
kolesnikov@gatech.edu

Mike Rosulek

Oregon State University
rosulekm@eecs.oregonstate.edu

Errata (in Reverse Chronological Order)

Last update: September 8, 2020

8 September 2020

Fixed grammatical error in first sentence!

13 April 2020

Many corrections suggested by Weiran Liu and Shengchao Ding. The substantive ones are:

- Figure 2.3: The notation C should be replaced by C .
- Figure 3.1 relabeled as Table 3.1 (and references fixed).
- p. 41: “Generalization to more than two parties. ... where n players P_1, P_2, \dots, P_n evaluate a boolean circuit F ” should be C .
- p. 53: “by setting both subshares of the first wire to a random string $R_1 \in_R D$ ” should be $R_1 \in_R \mathcal{D}_S$.
- p. 54, Section 3.6, last paragraph: “Then P_1 transfers to P_2 active wires on the input labels” should be “Then P_1 transfers to P_2 active labels on the input wires.”
- p. 61, Section 3.8.1: Replaced Alice and Bob with P_1 and P_2 .
- Figure 4.1: In 3(a), the notation $p_a \oplus p_b$ should be $p_a^0 \oplus p_b^0$.
- Figure 4.1: The notation R (in 3(b)) should be replaced by Δ .
- p. 71: to obtain either c_0 (should be c^0) (false, when $b = b_0$ (should be b^0)) or $c^1 = c^0 \oplus \Delta$ (true, when $b^1 = b^0 \oplus \Delta$ (should be $b = b^1$)). Similar problem in the line before ($c_0 \oplus b_0$ should be $c^0 \oplus b^0$).
- p. 88, Figure 5.1, caption: A single array access requiring n (should be N) multiplexers.
- p. 90, above Other Oblivious Data Structures: the total circuit size for k operations is $O(k \log n)$ (should be $O(k \log N)$).

- p. 95, first paragraph: "...could be implemented with less than 0.0001 probability of overflow for $\delta = 32$ " should be "for a bucket size of 32".
- p. 99, first paragraph: The missing close parentheses should be after "function" earlier in this sentence, $y_p^x = P_p^{\alpha,\beta}(x)$ (party p 's share output of the function), and $t_p^x = (x = \alpha)$ (a share of 1 if $x = \alpha$, otherwise a share of 0).
- Figure 6.1: should be Table 6.1.
- p. 109, first paragraph: "circuits agree, or by recovering P_2 's" should be P_1 's.
- p. 130: P_2 computes s_3 should be s_2 .
- p. 136, paragraph 2: $x_i =_{j \in \{1..i\}} x_i^j$ should be $j \in \{1..\sigma\}$.
- p. 136, last paragraph: "Then, instead of P_2 just sending the keys associated with its input, it sends the appropriate decommitments." should be P_1 .

23 June 2019

- Footnote 1 on Page 34 (Patricia Thaine): "will reveal x to P_1 " should be "will reveal x to P_2 ".
- Section 4.1.2 (p. 67, bottom) (Patricia Thaine): The share reconstruction description didn't include the semantic indexes. To clarify, it should be:

The share reconstruction procedure on input sh_{1i}, sh_{2i} , outputs $sh_{1i} \oplus sh_{2i} = s_i$.

- Section 6.2 (p. 109) (Patricia Thaine):

"It follows that the parties must always perform the second phase, even when P_1 is honest."

should be

"It follows that the parties must always perform the second phase, even when P_1 is caught cheating."

- Section 6.5.1 (p. 113-114) (Patricia Thaine): The given wording could be interpreted ambiguously,

“In other words, the ZK proof should prevent parties from running π honestly, but with different inputs in different rounds.”

Replaced with:

“In other words, the ZK proof should prevent parties from running π with different inputs in different rounds.”

10 July 2019

- Fixes to notation in Section 4.1 (the GESS construction) to avoid confusion in the Δ notation. (Shengchao Ding)

23 Aug 2019

- Section 4.1.3, p. 71, line 2-3 (Shengchao Ding): “when v_a is false, $v_c = v_b$ ” should be “when v_a is true, $v_c = v_b$ ”
- Section 4.2.2, several instances (Shengchao Ding): “CMBC-GC” should be “CBMC-GC”

2 October 2019

- Figure 3.4 (BMR Multi-Party GC Generation) (Kelong Cong): line 23 of the figure has $w_{c,1}^0$, but it should be $w_{c,1}^1$.