Lecture 13

Tuesday, 10 January 2023 08:22

multi-party computation

two-party computation → possible with fully homomorphic encryption

example:

$a_0 = a \oplus a_1 \oplus a_2 = 0$

in $\mathbb{F}_2$, $+ \equiv \oplus$

$\cdot \equiv \wedge$

$$a \wedge b = (a_0 \oplus a_1 \oplus a_2) \wedge (b_0 \oplus b_1 \oplus b_2)$$

$$= (a_0 + a_1 + a_2) \cdot (b_0 + b_1 + b_2)$$

$$= a_0 b_0 + a_0 b_1 + a_0 b_2 + \dots$$

only two parties needed per term!

---

without definitions, it is impossible to determine if something is broken

MPC:

completeness: all parties honest ⇒ result correct

fairness: all or no parties receive result — requires honest majority

soundness: scheme is 'secure'

simulation-based security notion

adversary may corrupt subset of all parties

and can only learn their in-/outputs and some leakage, defined for the functionality

e.g. length of message

check whether simulated transcript (using ideal functionality) is indistinguishable from real transcript

---

bits a and b

learn $a \wedge b$, nothing else

give a to Bob, compute $a \wedge b$

secure against active attacker, because choosing $b = 1$

gives $a \wedge b = a$

and hence does not 'leak' anything