XVII. Secure Function Evaluation

XVII.1. Beispiel: Datingproblem

Alice und Bob haben ein Rendezvous, vermittelt von Datingagentur. Wie entscheiden, ob man sich wieder trifft? (peinlich, falls nur einer will...)

XVII.1.1. Lösung: Secure AND

jeder gibt geheimes Bit ein und lernt nur das AND-Ergebnis

Realisierung

- Input Alice: Bit a
- Input Bob: Bit b

1. Alice generiert Signaturkey x und sendet Verifikationskey y an Bob
2. Alice generiert (e, d, N) und sendet (e, N) an Bob
3. Bob wählt s, berechnet $r_b := Enc(s)$, $r_{1-b} := N - b$ und sendet (r_0, r_1) an Alice
4. Alice sendet $c_0 = sign(0) + Dec(r_0)$ und $c_1 = sign(a + Dec(r_1))$ an Bob
5. Bob berechnet $sign(ab) = c_b - s$ und teilt Alice das Ergebnis mit

Wofür Signieren? → sonst kann Bob immer $b = 1$ wählen und Alice über das Ergebnis belügen (und a herausfinden).
Wenn Alice nur eine Signatur richtig berechnet, hängt es von b ab, ob das Protokoll regulär durchgeht. → Lösung: ZK-Beweis

Achtung: Signatur darf keinen Subliminal Channel haben

XVII.2. allgemeine Secure Function Evaluation

XVII.2.1. Baustein: „oblivious transfer“ (OT)

- Input Alice: Strings a_0, a_1
- Input Bob: Bit b
- Output Alice: nichts
- Output Bob: a_b
- Alice lernt nichts über b
- Bob lernt nichts über a_0 oder nichts über a_1

XVII.2.2. Realisierung

1. Alice generiert (e, d, N) und sendet (e, N) an Bob
2. Bob wählt s, berechnet $r_b = Enc(s)$, $r_{1-b} = N - b$ und sendet (r_0, r_1) an Alice
3. Alice sendet $c_0 = a_0 + Dec(r_0)$ und $c_1 = a_1 + Dec(r_1)$ an Bob
4. Bob berechnet $a_b = c_b - s$

XVII.2.3. Realisierung mit Funktion f

hier ist der Input von Bob ein String b

- Alice stellt Wertetabelle F für $f(a, \cdot)$ auf
- Für jedes Bit von b sendet Alice zwei Zufallswerte k_0, k_1 via OT an Bob; es bezeichne (k_0^i, k_1^i) das zu b_i gehörige Tupel. Bob lernt jeweils $k_{b_i}^i$.
- Alice verschlüsselt F zeilenweise; die zu b zugehörige Zeile von F ist verschlüsselt mit dem XOR über $k_{b_i}^i$.
- Alice sendet die verschlüsselten Zeilen an Bob.
- Bob kann genau die zu b gehörige Zeile entschlüsseln und lernt $f(a, b)$.

Probleme:

1. Wie sicherstellen, dass Alice die richtige Funktion f codiert? → ZK-Beweis
2. Wie sicherstellen, dass Bob über das Ergebnis nicht lügen kann? → Signaturen
3. Tabelle F ist exponentiell groß in $|b|$ → „Yao's Garbled Circuits“

XVII.2.4. Yao's Garbled Circuits

- Zerlege Schaltkreise von f in einzelne Gatter und wende obiges Protokoll auf jedes einzelne Gatter an.
- Bob darf keine Zwischenergebnisse lernen, deshalb werden Gatterergebnisse direkt durch entsprechende Schlüssel ersetzt, statt die Schlüssel via OT zu übertragen (natürlich nur für Gatter, deren Ergebnis nicht Teil der Ausgabe des Schaltkreises ist).
- Bob weiß nicht mehr, welche Zeile einer Tabelle für ein Gatter innerhalb des Schaltkreises er entschlüsseln muss; deshalb werden verschlüsselte Nullstrings an jede Tabellenzeile angehängt, Bob entschlüsselt immer die ganze Tabelle und verwirft „falsche“ Zeilen.
- Bob kann noch aus der Position der „richtigen“ Zeile einers Gatters die Eingangsbelegung erkennen; daher werden jeweils die Zeilen einer Tabelle zufällig permutiert.