Bezpečné výpočty viacerých účastníkov

Martin Stanek

2025

KI FMFI UK Bratislava

Úvod

- dva prístupy k bezpečným výpočtom:
  - plne homomorfné šifrovanie (FHE) a outsourcované výpočty
  - viacstranné výpočty (Secure Multiparty Computation, MPC)
- FHE: vysoká výpočtová zložitosť
- MPC: vysoká komunikačná zložitosť
  - generické vs. špecializované protokoly
  - rôzne modely, napr. komunikačné, hrozby (útočníci), bezpečnosť

Skupina účastníkov počíta spoločne funkciu, pričom si navzájom nedôverujú a chcú zachovať súkromie svojich vstupov.

Hlavné modely bezpečnosti

- dva hlavné modely z hľadiska správania sa účastníkov
- poločestní účastníci (semi-honest)
  - účastníci dodržujú protokol tak, ako je predpísaný
  - po skončení sa snažia extrahovať dodatočnú informáciu z komunikácie
- nečestní účastníci (malicious)
  - účastníci nedodržiavajú protokol, s cieľom získať informáciu o ostatných vstupoch
- Poznámky:
  - spomeňme si na ZK (bezznalostnosť) pre čestného overovateľa vs. ZK pre ľubovoľného overovateľa
  - *dodatočná informácia* – okrem tej, ktorá sa dá vypočítať z vlastného vstupu a výstupu funkcie

Jednoduchý príklad – výpočet súčtu (priemeru)

- n účastníkov $P_1, \ldots, P_n$
- každý s vlastnou hodnotou m_i
- cieľ: vypočítať $\sum_{i=1}^{n} m_i$ bez prezradenia čohokoľvek ďalšieho
- predpokladajme, že $\sum_{i=1}^{n} m_i < M$
  - všetky + a − operácie sú počítané mod M
- pri výpočte priemeru stačí deliť n

1. P_1 zvolí $r \in_R \mathbb{Z}_M$ a pošle P_2 hodnotu $m_1 + r$
2. postupne: P_i pripočíta m_i k prijatej hodnote a pošle P_{i+1}
3. P_1 dostane od P_n: $r + \sum_{i=1}^{n} m_i$ odpočíta r a oznámi výsledok

- predpokladáme poločestných účastníkov
  - protokol bezpečný voči jednotlivcovi
  - protokol nie je bezpečný voči dvojici (napr. P_2 a P_4 vedia zistiť m_3)

Oblivious Transfer (OT)

- základný protokol pre konštrukciu všeobecných MPC (OT $\Longrightarrow$ MPC)
- účastníci: odosielateľ S, príjemca R

1–2 OT

- S má dvojicu správ m_0, m_1
- R má vstup $b \in \{0, 1\}$
- ciele 1–2 OT protokolu:
  - výstup: R sa dozvie m_b
  - súkromie S: R sa nedozvie nič o m_{1-b}
  - súkromie R: S sa nedozvie nič o b

- zovšeobecnenie **1–t OT**:
  - S má t správ: $m_1, \ldots, m_t$
  - R má vstup $b \in \{1, \ldots, t\}$
  - ciele ako pre 1-2 OT, zovšeobecnené pre t správ

1-2 OT pre poločestných účastníkov

- R zvolí
  - inštanciu asymetrického šifrovania s verejným kľúčom pk
  - náhodne zvolí verejný kľúč pk′ (v schéme musí byť verejný kľúč takto voliteľný)
- pri poločestných účastníkoch sú ciele protokolu naplnené
  - ak R nedodrží protokol pri voľbe pk′, dozvie sa aj m_{1-b}
- ľahko zovšeobecniť na 1-t OT

1. $R \to S$: $(\mathrm{pk}, \mathrm{pk}')$ ak $b = 0$, resp. $(\mathrm{pk}', \mathrm{pk})$ ak $b = 1$
2. $S \to R$: $E_{\mathrm{pk}_0}(m_0), E_{\mathrm{pk}_1}(m_1)$, kde $(\mathrm{pk}_0, \mathrm{pk}_1)$ je prijatá dvojica kľúčov
3. R dešifruje m_b s použitím pk

Ako zamedziť podvodnej voľbe pk′?

ElGamalova šifrovacia schéma

- grupa G prvočísleného rádu q
- generátor $g \in G$
- $H : G \to \{0,1\}^l$, hašovacia funkcia (RO)
- súkromný kľúč $x \in_R \mathbb{Z}_q$
- verejný kľúč $y = g^x$
- otvorený text $m \in \{0,1\}^l$
- šifrovanie:
  $(g^k, H(y^k) \oplus m)$, kde $k \in_R \mathbb{Z}_q$
- dešifrovanie:
  $(H(y^k) \oplus m) \oplus H((g^k)^x) = m$

1–2 OT protokol

1. $S \to R$: $c \in_R G$
2. $R \to S$: (y_0, y_1), kde
   $y_b = g^x$ pre $x \in_R \mathbb{Z}_q$ a $y_{1-b} = c y_b^{-1}$
3. S overí, že $y_0 y_1 = c$
   (ukončí protokol ak rovnosť neplatí)
4. $S \to R$: $(g^{k_0}, H(y_0^{k_0}) \oplus m_0)$,
   $(g^{k_1}, H(y_1^{k_1}) \oplus m_1)$, kde $k_0, k_1 \in_R \mathbb{Z}_q$
5. R dešifruje m_b pomocou x

1–t OT založený na „trapdoor“ permutácii

- trapdoor permutácia – napr. RSA (trapdoor ≈ súkromný kľúč)
- opäť predpokladáme poločestných účastníkov
- existuje veľa konštrukcií OT protokolov
  - optimalizácia výpočtovej a komunikačnej zložitosti
  - redukcia počtu asymetrických operácií pri veľkom počte OT protokolov

Protokol (správy $m_1, \ldots, m_t \in \mathbb{Z}_n$):

1. $S \to R$: (n, e) verejný RSA kľúč
2. $R \to S$: $\{c_i\}_{i=1}^{t}$, kde $c_b = E(\alpha)$ pre $\alpha \in_R \mathbb{Z}_n^*$; ostatné $c_i \in_R \mathbb{Z}_n$
3. $S \to R$: $\{D(c_i) \cdot m_i \bmod n\}_{i=1}^{t}$
4. R vypočíta $(D(c_b) \cdot m_b) \cdot \alpha^{-1} = \alpha \cdot m_b \cdot \alpha^{-1} = m_b \pmod n$

2PC s malým definičným oborom

- $f(x, y)$, kde $x \in X$, $y \in Y$ a $|X|$ aj $|Y|$ sú malé
  - f ako look-up tabuľka T s $|X| \cdot |Y|$ riadkami
  - neškálovateľné
- vstupy účastníkov:
  - A má vstup $\boldsymbol{x}$, B má vstup $\boldsymbol{y}$
- E – vhodná symetrická šifra

Protokol:

1. A zvolí náhodný, nezávislý kľúč pre každé $x \in X$ (označme k_x) a pre každé $y \in Y$ (označme k_y)
2. A zašifruje každý riadok T: $T_{x,y} \mapsto E_{k_x,k_y}(T_{x,y})$
3. $A \to B$: $k_{\boldsymbol{x}}$, náhodne permutovaná zašifrovaná tabuľka T
4. $A \leftrightarrow B$: 1–$|Y|$ OT, prenos hodnoty $k_{\boldsymbol{y}}$
5. B má $k_{\boldsymbol{x}}$ a $k_{\boldsymbol{y}}$ a môže dešifrovať $T_{\boldsymbol{x},\boldsymbol{y}}$

2PC s malým definičným oborom (pokr.)

B: ktorý riadok riadok je ten správny (náhodná permutácia)?

idea 1 – pred šifrovaním pridať napr. konštantnú výplň

- potrebné dešifrovať riadky pokiaľ nenájdeme správnu výplň
- dopad na výpočtovú zložitosť (v priemere dešifrujeme polovicu riadkov)
- dopad na komunikačnú zložitosť (predĺženie ŠT)

idea ?
A pošle len tie riadky T, kde $x = \boldsymbol{x}$

idea 2 (point-and-permute)

- ku k_x (k_y) pridáme náhodnú permutáciu $\{1, \ldots, |X|\}$ (resp. $\{1, \ldots, |Y|\}$)
- určená náhodná permutácia look-up tabuľky (pre x a y)
- po prenose $k_{\boldsymbol{x}}$, resp. OT prenose $k_{\boldsymbol{y}}$ vie B určiť riadok, ktorý dešifruje

- neformálna úvaha:
  - A a B vypočítajú $f(\boldsymbol{x}, \boldsymbol{y})$
  - A sa nedozvie nič o $\boldsymbol{y}$ – OT protokol
  - B sa nedozvie nič o $\boldsymbol{x}$ – náhodná permutácia šifrovaných riadkov T

Garbled Circuits protokol

- Yao (1986)
- všeobecný protokol pre dvoch poločestných účastníkov (A, B)
- funkcia reprezentovaná ako booleovský obvod (daný príslušnými hradlami)
  - booleovské hradlá majú obmedzený definičný obor
  - protokol postupne vyhodnotí obvod
- veľkosť obvodu má vplyv na zložitosť:
  - sčítanie, porovnanie, násobenie, AES, nepriama adresácia $A[i]$, podmienky, ...
- bez ujmy na všeobecnosti $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^*$
  - vstupy A: $x_1, \ldots, x_n$, vstupy B: $y_1, \ldots, y_n$
- A vytvorí *skomolenú* verziu obvodu pre f (teda A zvolí kľúče na hranách obvodu)

Skomolenie obvodu

- každý prepoj/vodič/hrana w (vstupný, interný, výstupný):
  - priradená dvojica náhodných kľúčov k_w^0, k_w^1
- každé hradlo (AND, XOR a pod.) je skomolené takto:
  - nech k_a^0, k_a^1, resp. k_b^0, k_b^1 sú kľúče zodpovedajúce vstupom hradla
  - nech k_c^0, k_c^1 sú kľúče zodpovedajúce výstupu hradla

a	b	AND
0	0	0
0	1	0
1	0	0
1	1	1

$\Longrightarrow$

a	b	AND
k_a^0	k_b^0	$E_{k_a^0}(E_{k_b^0}(k_c^0))$
k_a^0	k_b^1	$E_{k_a^0}(E_{k_b^1}(k_c^0))$
k_a^1	k_b^0	$E_{k_a^1}(E_{k_b^0}(k_c^0))$
k_a^1	k_b^1	$E_{k_a^1}(E_{k_b^1}(k_c^1))$

Protokol

1. $A \to B$: kľúče pre vlastné vstupné bity $k_1^{x_1}$, ..., $k_n^{x_n}$
   $A \to B$: permutované tabuľky šifrových textov skomolených hradiel
   – použijúc výplň alebo point-and-permute techniku
   – pri g hradlách $4g$ šifrových textov
2. $A \leftrightarrow B$ prenos kľúčov zodpovedajúcich vstupným bitom B
   – n krát 1–2 OT (aj paralelne), pre každú dvojicu kľúčov zodp. vstupnému bitu B
3. B vyhodnotí booleovský obvod
   – B má kľúče zodpovedajúce všetkým vstupným bitom a skomolené hradlá
   – B získa kľúče zodpovedajúce výstupným bitom obvodu
4. A prezradí B bity, ktoré zodpovedajú týmto kľúčom

Poznámky

- intuícia, prečo by to malo fungovať
  - korektnosť v modeli s poločestnými účastníkmi zrejmá
  - A sa nedozvie vstupy B – OT protokol
  - B sa nedozvie vstupy A – kľúče sú nezávislé na bitoch, permutácia skomolených hradiel
  - potrebná aj špecifická bezpečnosť dvojitého šifrovania
- konštantný počet kôl – podľa voľby OT protokolu 2 alebo 3 kolá
- rôzne optimalizácie – zamerané najmä na komunikačnú zložitosť
  - počet AND hradiel v obvode
  - free XOR (špecifické riešenie XOR hradiel) a ďalšie

MPC pre viacerých poločestných účastníkov

- predchádzajúce konštrukcie len pre 2PC
  - zovšeobecnenie existuje, napr. BMR protokol (Beaver-Micali-Rogaway)
- zovšeobecnenie pre n účastníkov
  - stále predpokladáme poločestných účastníkov
  - booleovský obvod (bez ujmy na všeobecnosti hradlá NOT, XOR, AND)
  - GMW a viaceré ďalšie konštrukcie – aj pre aritmetické obvody s hradlami $+, \cdot$
- GMW protokol (Goldreich, Micali, Wigderson, 1987)
  - účastníci: $P_1, \ldots, P_m$
  - základná myšlienka: hodnota na každej hrane obvodu bude distribuovaná v schéme spoločného tajomstva

- ukážme najskôr prípad $m = 2$ a potom zovšeobecnenie
- vstupy P_1: $x_1, \ldots, x_n \in \{0,1\}$; P_2: $y_1, \ldots, y_n \in \{0,1\}$
- počítame $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^*$

Protokol:

1. každý vstup P_1, resp. P_2 rozdelí v $(2,2)$-prahovej schéme spoločného tajomstva:
   $x_i = x_i^1 \oplus x_i^2$, resp. $y_i = y_i^1 \oplus y_i^2$, kde $x_i^1, y_i^1 \in_R \{0,1\}$
2. $P_1 \to P_2$: $x_1^2, \ldots, x_n^2$
   $P_2 \to P_1$: $y_1^1, \ldots, y_n^1$
3. postupné vyhodnotenie hradiel booleovského obvodu (pozri ďalej)
   - schémy spoločného tajomstva pre všetky hrany obvodu
4. P_1, P_2 si pošlú podiely pre výstupy obvodu

GMW – hradlá NOT a XOR

- $a = a^1 \oplus a^2, b = b^1 \oplus b^2$
- vstupy P_1: a^1, b^1; P_2: a^2, b^2
- výstup: $c = c^1 \oplus c^2$
- **NOT** (nevyžaduje žiadnu interakciu)
  - $c = \text{NOT } a$
  - P_1: $c^1 = 1 \oplus a^1$
  - P_2: $c^2 = a^2$
  - triviálne:
    $c^1 \oplus c^2 = 1 \oplus a^1 \oplus a^2 = 1 \oplus a = \text{NOT } a$
- **XOR** (nevyžaduje žiadnu interakciu)
  - $c = a \text{ XOR } b$
  - P_1: $c^1 = a^1 \oplus b^1$
  - P_2: $c^2 = a^2 \oplus b^2$
  - triviálne:
    $c^1 \oplus c^2 = (a^1 \oplus b^1) \oplus (a^2 \oplus b^2) = a \oplus b$

GMW – hradlo AND

- **AND**: $c = a \text{ AND } b$
- P_1 vytvorí potenciálne podiely, ktoré P_2 má v závislosti na jeho podieloch:
  - $S(a^2, b^2) = (a^1 \oplus a^2) \wedge (b^1 \oplus b^2)$
  - zvolí si svoj podiel $c^1 \in_R \{0, 1\}$
- 1–4 OT protokol na prenos *správneho* podielu hodnoty $S(a^2, b^2)$:
  - $c^1 \oplus S(0,0), c^1 \oplus S(0,1), c^1 \oplus S(1,0), c^1 \oplus S(1,1)$
- intuitívny argument pre zachovanie súkromia vstupov
  - úvodná distribúcia v schéme spoločného tajomstva
  - NOT, XOR – nevyžadujú žiadnu komunikáciu
  - AND – vlastnosti OT protokolu, pričom len P_2 dostane nejakú informáciu

GMW pre väčší počet účastníkov

- $P_1, \ldots, P_m$ $(m \geq 2)$
- úvodná distribúcia podielov:
  - $x_i = x^1 \oplus x^2 \oplus \ldots \oplus x^m$, kde $x^2, \ldots, x^m \in_R \{0, 1\}$
  - (m, m)-prahová prístupová štruktúra, perfektná schéma
- výpočet hradiel a vzájomné oznámenie podielov výstupov obvodu
- **NOT**: P_1 invertuje svoj podiel
- **XOR**: sčítanie podielov ako predtým – P_i: $c^i = a^i \oplus b^i$

GMW pre väčší počet účastníkov (pokr.)

AND:

$$\left(a^1 \oplus \ldots \oplus a^m\right) \wedge \left(b^1 \oplus \ldots \oplus b^m\right) = \bigoplus_{i=1}^{m} a^i b^i \quad \oplus \quad \bigoplus_{i \neq j} a^i b^j$$

- $a^i b^i$ vie vypočítať každý P_i samostatne
- výpočet jednotlivých $a^i b^j$:
  - P_i a P_j počítajú AND ako v protokole pre 2 účastníkov
  - 1–4 OT, na konci majú obaja podiel na tomto súčine ($c^i_{i,j}$ a $c^j_{i,j}$)
  - $c^i_{i,j} \oplus c^j_{i,j} = a^i b^j$
- podiel P_i na výstupe AND hradla je súčtom všetkých podielov:

$$c^i = a^i b^i \oplus \bigoplus_{j \neq i} c^i_{i,j}$$

Nečestní účastníci

- základná myšlienka:
  - protokol pre poločestných účastníkov + bezznalostné dôkazy dodržiavania protokolu ⇒ protokol pre nečestných účastníkov
- GMW kompilátor
  - existujú aj iné kompilátory (efektívnosť, dodatočné vlastnosti, špecializované na konkrétne typy protokolov)
  - určený pre ľubovoľný protokol pre poločestných účastníkov
- nečestní účastníci:
  - nekorektná konštrukcia podielov, nesprávne vstupy v OT protokole, nesprávne podiely výstupov obvodu, …

Kompilátor pre deterministický protokol

- bezpečný MPC protokol nebude deterministický (!)
- označenia:
  - $C(x, r)$ – záväzková (commitment) schéma pre x a náhodný reťazec r
  - π_i – program pre P_i v protokole
  - T_i^s – kompletná komunikácia P_i (prijaté a poslané správy) až po krok s (vrátane)
- P_i na začiatku vytvorí a pošle záväzok (commitment) svojho vstupu x_i: $C(x_i, r_i)$
- definujme jazyk $L_i = \{T_i^s \mid \exists x_i, r_i : T_i^s \in \pi_i(x_i, r_i)\}$
  - teda všetky správy poslané P_i sú výstupom programu π_i na vstupoch x_i, r_i a správach prijatých po krok s
  - $L_i \in \mathbf{NP}$, stačí si *tipnúť* x_i, r_i a deterministicky polynomiálne overiť
  - každý jazyk z **NP** má bezznalostný dôkaz (existuje ZK IDS), aj ho vieme skonštruovať

Kompilátor pre deterministický protokol (pokr.)

- P_i posiela správu v kroku s a zároveň dokáže, že $T_i^s \in L_i$
- overovateľ ZK dôkazu musí poznať prijaté/odoslané správy
  - triviálne pre dvoch účastníkov
  - pre $m > 2$ potrebujeme bezpečný broadcast
- P_i dokáže (ZK) znalosť hodnoty v záväzku (proof of knowledge)
  - umožňuje extrakciu vstupu pri simulácii
  - znemožňuje vytvorenie závislého záväzku nečestným účastníkom (bez znalosti hodnoty)

GMW kompilátor

- kompilátor aj pre protokoly s náhodnosťou (nie len deterministické)
  - napr. v GMW: konštrukcia podielov, OT
- problémy, ktoré je potrebné vyriešiť (prinútiť účastníkov používať čestne náhodné bity):
  - správy a výpočet v π závisia na obsahu náhodnej pásky
  - znalosť náhodnej pásky môže mať dopad na bezpečnosť (súkromie vstupov)
  - náhodná páska je uniformná
- tri základné fázy:
  1. Záväzky vstupov (ako predtým $C(x_i, r_i)$ pre náhodné r_i a ZK dôkaz znalosti x_i, r_i)
  2. *Hádzanie* mincou (obsah náhodnej pásky)
  3. Emulácia protokolu

GMW kompilátor – hádzanie mincou

- cieľ je mať uniformne distribuovanú náhodnú pásku pre každého účastníka, k obsahu ktorej je zaviazaný
  - teda nemôže klamať o obsahu pásky
  - ostatní účastníci nevedia o obsahu pásky nič, ale majú k dispozícii záväzok
- (neformálne, idea) určenie náhodnej pásky v prípade $m = 2$
  1. P_1 zvolí $b_1, b_2' \in \{0,1\}^n$
     $P_1 \to P_2$: $C(b_1, r_1), b_2'$, záväzok pre b_1
  2. P_2 zvolí $b_2, b_1' \in \{0,1\}^n$
     $P_2 \to P_1$: $C(b_2, r_2), b_1'$, záväzok pre b_2
  3. páska pre P_1: $b_1 \oplus b_1'$; páska pre P_2: $b_2 \oplus b_2'$
- ZK dôkazy potom zohľadňujú záväzok b_i a známu hodnotu b_i' a zabezpečujú, že π_i beží s náhodnou páskou $b_i \oplus b_i'$

GMW kompilátor – emulácia protokolu

- podobne ako v deterministickom prípade
- dodatočným argumentom π_i je náhodnosť daná súčtom z časti hádzanie mincou (prípad $m = 2$)
  - väzba na záväzok a *doplnkovú* hodnotu je daná tým, že tieto hodnoty sú súčasťou transkiptu
- P_i dokazuje konzistentnosť svojho výstupu ZK IDS voči každému účastníkovi

Poznámky

- GMW kompilátor je všeobecný, ale veľmi *drahý* (výpočtovo, komunikačne), napriek polynomiálnej zložitosti
  - ZK dôkazy, generovanie náhodnosti pre m účastníkov, OT protokoly
- iné kompilátory, optimalizácie, špecializované protokoly pre konkrétnu funkčnosť
- pre niektoré aplikácie sú generické konštrukcie s optimalizáciami
  - dostatočné pre praktické použitie
  - efektívnejšie ako špecializované protokoly
- príklady praktického použitia MPC:
  - rozdiely v mzdách v oblasti Bostonu (2017)
  - konverzia reklama → predaj (2017)
  - skúmanie hypotéz predčasného ukončenia IT štúdia v Estónsku (2016)
- prehľadový článok: Y. Lindell: *Secure Multiparty Computation (MPC)* (2020)

PSI pre poločestných účastníkov

- PSI (private set intersection)
  - obmedzíme sa na 2 účastníkov
  - predpokladajme, že obaja majú rovnaký počet prvkov
  - $P_1: X = \{x_1, \ldots, x_n\}$; $P_2: Y = \{y_1, \ldots, y_n\}$
  - cieľom je vypočítať $X \cap Y$
- konverzia reklamy, identifikácia útočiacich IP adries a pod.
- naivné riešenie:
  1. $P_1 \rightarrow P_2: H(x_1), \ldots, H(x_n)$
  2. P_2 vypočíta $H(y_1), \ldots, H(y_n)$ a teda vie určiť prienik a oznámiť výsledok P_1
- problém: (potenciálne) malý priestor, z ktorého sú vstupné prvky a možnosť testovať príslušnosť do množiny vďaka odtlačku

PSI s využitím OPRF

- spomínate si na OPRF (Oblivious Pseudorandom Function)?
  - použitá na konštrukciu PAKE protokolu odolnému voči predvýpočtom
  - pseudonáhodná funkcia $F_k(x)$
  - OPRF je protokol s dvoma účastníkmi C (so vstupom x) a S (vstup k)
  - C sa dozvie $F_k(x)$ ale nič navyše
  - S sa nedozvie nič (napr. nič o x)

- postup:
  1. P_1 zvolí k
  2. $P_1 \leftrightarrow P_2$: n krát OPRF $F_k(y_i)$, pre $i = 1, \ldots, n$
  3. $P_1 \rightarrow P_2$: $F_k(x_1), \ldots, F_k(x_n)$
  4. P_2 vie určiť prienik a (ak je to potrebné) oznámi výsledok P_1
- bezpečnosť pre poločestných účastníkov sa opiera o vlastnosti OPRF
- v prípade nečestných účastníkov je potrebný komplikovanejší protokol (napr. využívajúci cuckoo hashing)

Konkrétny príklad PSI (RO, zovšeobecnený DDH predpoklad)

- $(G, \cdot)$ – grupa prvočíselného rádu q
- $H : \{0,1\}^* \to G$ (hašovacia funkcia, RO)

1. P_1 zvolí $\alpha \in_R \mathbb{Z}_q$
   P_1 vypočíta $a_i = H(x_i)^{\alpha}$, pre $i = 1, \ldots, n$
   $P_1 \to P_2$: $a_1, \ldots, a_n$
2. P_2 zvolí $\beta \in_R \mathbb{Z}_q$
   P_2 vypočíta $b_i = H(y_i)^{\beta}$, pre $i = 1, \ldots, n$
   P_2 vypočíta $c_i = a_i^{\beta}$, pre $i = 1, \ldots, n$
   $P_2 \to P_1$: $(b_1, \ldots, b_n), (c_1, \ldots, c_n)$
3. P_1 vypočíta $d_i = b_i^{\alpha} = H(y_i)^{\alpha\beta}$,
   pre $i = 1, \ldots, n$
   P_1 určí prvky x_i prieniku: $(\exists d_j)\ c_i = d_j$

- korektnosť
  - ak $x_i = y_j$, tak $c_i = d_j$
    vtedy máme $H(x_i)^{\alpha\beta} = H(y_i)^{\alpha\beta}$
  - ak $x_i \neq y_j$, tak $c_i \neq d_j$ s vysokou pravdepodobnosťou, lebo H je RO
- súkromie účastníkov sa opiera o zovšeobecnený DDH predpoklad
- vhodné pre výpočet prienikov „malých“ množín