10/02/2013

Sublinear secure computation?

- All protocols thus far were circuit-based
  → any circuit for a non-trivial function has linear size

- Any secure computation protocol requires at least linear time
  → if a party does not touch every bit of input, this leaks information

Idea:

- Move to a setting where parties evaluate some function l times

- Hope for complexity $O(n) + o(nl)$
  $\Rightarrow$ amortized complexity $O(n/l) + o(n)$
  $\Rightarrow$ l large $\Rightarrow$ sublinear amortized complexity

- Or preprocessing (expensive)
  → online computation sublinear in n

Example: Binary-Search:

- init setup
- for each binary-search computation
  - each lookup for the binary-search algorithm is compiled into a sequence of actual lookups in the ORAM.

One-sided secure protocol: (privacy of clients)

initialization

Client — **Server (D)**

$\xleftarrow{D}$

$st, \hat{D} \leftarrow Init(D)$

$\xrightarrow{\hat{D}}$

---

operations

Client (i) — **Server ($\hat{D}$)**

st

run ORAM-compiled algorithm locally

whenever $\hat{D}[j]$ is needed, send j to the server $\xrightarrow{op, j, data}$ fetch $= \hat{D}[j]$

if (op = write) $\hat{D}[j]$ = data

$\xleftarrow{\hat{D}[j]}$

Two-sided secure protocol:

initialization

Client — **Server (D)**

$\xleftarrow{st_c}$ $\boxed{Init}$ $\xleftarrow{D}$; $\xrightarrow{\hat{D}, st_s}$

$(st = st_c \oplus st_s)$

---

operations

Client (i, st_c) — **Server ($\hat{D}, st_s$)**

$\xrightarrow{st_c}$ $\boxed{NextInstr}$ $\xleftarrow{st_s,\ fetched}$

$\xleftarrow{st'_c}$ $\boxed{NextInstr}$ $\xrightarrow{op, j, data, st'_s}$

server sets fetched $= \hat{D}[j]$

if (op = write) $\hat{D}[j]$ = data

$\xrightarrow{st_c}$ $\boxed{Done?}$ $\xleftarrow{st_s}$

$\xleftarrow{val/no}$ $\boxed{Done?}$ $\xrightarrow{yes/no}$

Improvement: run for fixed # of steps

* online: sublinear time

* secure for semi-honest model

Generic Protocol:

- protocol that can securely evaluate any function in some specified class of functions
  → e.g. finite automaton

Special-purpose / tailored protocol:

- designed specifically for some function f

Problem: Set Intersection:

each P_i has a set $S_i \subseteq \{0,1\}^\ell$

Parties want to compute $S_1 \cap S_2$

ENP '04:

- homomorphic encryption-based solution

observation:

Let $P(x) = \sum_i a_i x^i$

Given $[a_0], [a_1], \ldots, [a_\ell], y$

Can compute $[P(y)]$

observation:

$P_1(\{x_1, \ldots, x_n\})$ $\qquad$ $P_2(\{y_1, \ldots, y_n\})$

Encode its set as 0's of a polynomial Q.

i.e. $Q(x) = \prod_i (x - x_i)$

$$Q(x) = \sum_i a_i \cdot x^i$$

$$\xrightarrow{pk, [a_0], [a_1], \ldots, [a_n]}$$

compute $[Q(y_1)], \ldots, [Q(y_n)]$

$\Downarrow$

compute $[r_1 Q(y_1) + y_1]$

$\vdots$

$[r_1 Q(y_n) + y_n]$

$$\xleftarrow{[r_1 Q(y_1) + y_1], \ldots, [r_n Q(y_n) + y_n]}$$

decrypt...