ORAM Lower bound \cite{LarsenN18}

\[ \text{[GO '96]} \]

Client cpu \hspace{1cm} \begin{array}{c}
\text{(untrusted)} \\
\text{server}
\end{array}

\begin{array}{c}
\text{RAM} \\
\text{Word size:} \ W
\end{array}

\begin{array}{c}
\text{Eve} \\
\text{read/}
\text{write}
\end{array}

\begin{array}{c}
\text{f.} \\
\text{register/mem}
\end{array}

\text{word size:} \ R

\begin{itemize}
\item \( N \) accesses take \( \frac{M \cdot w}{N \cdot R} \) access to ORAM
\end{itemize}

\text{(bandwidth) overhead}

\text{Upper bounds}

\[ \text{[GO '96] poly log N} \]

\[ \text{[Stefanov, van Dijk, Shi, Chan, Fleischer, Ren, Yu, Devades '13]} \]

\[ O(\log N) \text{ but } W = O(\log^r N) \quad R = W \]

\[ \text{Path ORAM} \]

\[ \text{[PPRY '18] PanORAMa } O(\log N \log \log N) \quad W = O(\log N) = r \]

\[ \text{[AHLNPS '18] OptORAMa } O(\log N) \quad W = O(\log N) = r \]

\text{Lower bound}

\[ \text{[GO '96] } \Omega(\log N) \]

- "balls and bins": the algorithm can't read the contents
- statistically secure: unbounded adversary

GO'96, and many constructions followed are only computationally secure.

\[ \text{[Boyle, Naor '16] Is there an ORAM Lower bound?} \]

Thm. Suppose there is a circuit sorting \( n \) words with \( w \)-bits, with size \( o(n^w \log n) \), then there exists offline ORAM compiler with overhead \( o(\log N) \)

\text{(offline) ORAM lower bound } \Rightarrow \text{efficient sorting circuits } \n\times \text{balls-bins}
online ORAM?

[Laarhoven, Nielsen '18] Yes, there is an ORAM Lower Bound!

- \( \Omega(\log n) \) for online ORAM
- any algorithm
- computationally secure
- any block size \( w \)

\[ \Omega\left(\log\left(\frac{Nw}{m}\right)\right) \] \( r \): word size for client

\( m \): total memory bits for client

\( o(1) \) blocks, \( r \leq w \leq n^{0.3} \Rightarrow \log n \)

Array maintenance problem for dynamic array

\[
\begin{aligned}
&\text{Write} \ (i, \text{data}), \ \text{data} \in \{0,1\}^r \\
&\text{Read} \ (i)
\end{aligned}
\]

re-use data structure lower bounds

\( U \): updates \quad \( Q \): queries

Cell probe model

[Planar, Demaine '13]

Idea:

1) \( \text{ops: } w(1,1) \quad w(2,2) \quad r(1) \quad r(2) \)

"access prev. accessed cells" must happen \( N/2 \)

2) \( \text{ops: } w(0,0) \quad w(0,0) \quad r(0) \quad r(0) \)

also need \( N/2 \) accesses w.r.t. security breach easy

2) \( \text{ops: } w(1,1) \quad r(1) \quad w(2,2) \quad r(2) \)

\( N/4 \times N/4 \times N/4 \times N/4 \)

\( N/4 \): \( N/4 \) accesses \( N/4 \) accesses

Client
Oblivious Cell Probe

Complexity: amortized over $M$ operations
It probes in expectation over $v \in \{0,1\}^d$ uniform

Security: $y := (o_1, ..., o_M)$ op sequence
$A(y) := (A(o_1), ..., A(o_M))$ probe sequence

$y \neq 2 \quad A(y), A(z) \text{ distinguished w/ prob. } > \frac{1}{4}$

in poly time $\log |L| + \log |A| + w$

Correctness: fail prob. $< \frac{1}{4}$

Thm. $\exists y := (o_1, ..., o_M) \quad o_i \in U \forall i$ s.t.
assuming security holds, takes $\Omega(\log \log (N/r/m) \cdot r/m)$
(bandwidth overhead $\frac{M \cdot \log (N/r/m)}{M \cdot \log (N/r/m)}$)
for $r \leq m < N^{1/2}$

$y := w(0,0) \quad r(0) \quad w(0,0) \quad r(0) \quad \cdots$ $w(0,0) \quad r(0)$

$M = 2n$ operations

$\text{information transfer}$
\[ T(y) \] denote the tree for \( y \)
\[ T(z) \] denote the tree for \( z : = w(i_1, d, i_2) \cdots w(i_{dn}, d, i_{2n}) \)

Fix \( v \), depth \( d \).

Def \( P_v(y) \): # probes assigned to \( v \) in tree \( y \)

assuming \( < \frac{1}{4} \) fail prob. \( \forall \) in tree \( y \)

Lemma 1: \( \mathbb{E}(|P_v(y)|) = \Omega \left( \frac{nnr}{w2^d} \right) \) for \( d \leq \frac{1}{2} \log \frac{nr}{n} \)

then \( \mathbb{E}(|A(y)|) \geq \sum_v \mathbb{E}|P_v(y)| = \Omega \left( n \log(nr/m) \cdot n/w \right) \)

Take \( Z_v \) random op sequence in the form of \( z \)

Lemma 2: Assume \( < \frac{1}{2} \) fail prob. \( \exists \) universal constant \( c \)

\[ \mathbb{P}(P_v(Z_v) \geq Cnr/w2^d) \geq \frac{1}{2} \]

information transfer is large for random \( Z \)
on each node \( v \)

Consequence: \( \exists \) \( z \) s.t. lem 2 happens
then \( \mathbb{E}|P_v(y)| \geq \frac{1}{4} \frac{nr}{w2^d} \)

\( \text{o.w. } \mathbb{P}(P_v(y) \geq Cnr/w2^d) = \frac{1}{4} \) \( \text{ gap} \)
& \( \mathbb{P}(P_v(z) \geq Cnr/w2^d) \geq \frac{1}{2} \)

adversary can reconstruct \( T_a \) given \( a \in \{y, z\} \)
and observe transfer on nodes

\( \text{distinguishes } y, z \) w/ prob. \( > \frac{1}{4} \)

Proof of Lemma 2:
Assume \( \text{o.w. } \mathbb{P}(|P_v(Z_v)| \geq 100 \cdot \frac{nr}{w2^d}) < \frac{1}{2} \)
Encoding argument of data. Let subtree of \( v \) be \( d_0, \ldots, d_{n+\log_2(n+1)-1} \) of entropy, Shannon's source coding thm.

Any encoding must use \( nr/2^{d+1} \) bits in expectation conditioned on \( R \).

**Encode**

Now, if \( P_v(Z_v) \geq \) or error is large

- Encode directly

- If \( P_v(Z_v) < \) and error is small, write 1

    - Use \( D \) write down all contents related to \( P_v(Z_v) \)
    - and the state of \( D \) before it.
    - Also all errors

**Decode**

Bob can simulate the entire tree and retrieve all reads, thus all \( d_0, \ldots, d_{n+\log_2(n+1)-1} \).
Analysis

\[ c < 1 \]

\[ 1 + \frac{\frac{3}{4} \cdot \frac{\ln \gamma}{\ln \delta}}{\ln \gamma} + \frac{1}{\frac{1}{4} \cdot \ln \frac{\ln \gamma}{\ln \delta}} < \frac{\ln \gamma}{\ln \delta} \]

- online ORAM \( \Omega \left( \log \left( \frac{\ln \gamma}{m} \right) \right) \)