Tight Time-Space Lower Bounds for Finding Multiple Collision Pairs and Their Applications

Itai Dinur

Ben-Gurion University, Israel

Eurocrypt 2020

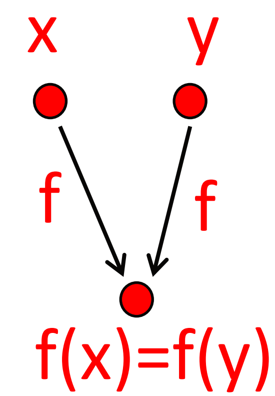
The Birthday Problem

- Let $[N] = \{0,1,\ldots,N-1\}$

- Given oracle access to **random function** $f:[N]\to[N]$:
  Goal: output colliding pair: (x,y), $x \neq y$ such that $f(x) = f(y)$

- Can be done in time (queries) T such that $T^2 \approx N$

- Tight (birthday bound)



Generalization of Birthday Problem

- Given access to **random function** $f:[N]\text{->}[N]$, parameter C:
  Goal: output C district colliding pairs $(x_1,y_1),\ldots,(x_C,y_C)$
- **Variant 2**: for random $f_1,f_2 : [N]\text{->}[N]$, parameter C:
  Goal: output C colliding pairs $(x_1,y_1),\ldots,(x_C,y_C) : f_1(x_i) = f_2(y_i)$
  - Variants **essentially equivalent**
- Can be done in time T such that $T^2\approx C\cdot N$
- Tight (generalized birthday bound)

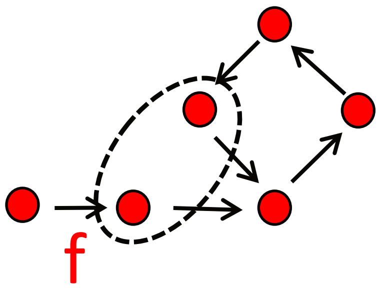
The Collision Pair Search Problem

- Given **random function** f:[N]->[N], parameter C:
  Goal: **output** C district colliding pairs $(x_1,y_1),\ldots,(x_C,y_C)$
- Can be done in time T such that $T^2 \approx C \cdot N$ (**tight**)
- What if **space** restricted to S bits?
- For $S \approx C$, parallel collision search (**PCS**) [vOW96']) gives $T^2 \approx C \cdot N$ (optimal)
- What if $S << C$?

The Collision Pair Search Problem

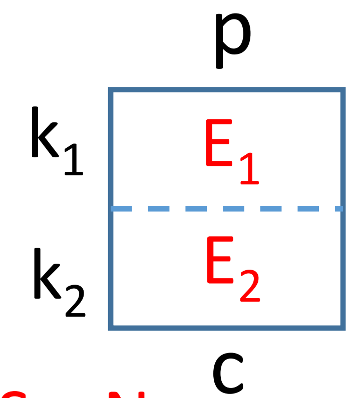
- For any S, PCS variant gives $T^2 \cdot S \approx C^2 \cdot N$
  - $S \approx C$ gives $T^2 \approx C \cdot N$
- E.g., for $S \approx 1$, $C \approx N$: $T \approx N^{1.5}$
  (generalized birthday bound is $T \approx N$)
  - "Memoryless" cycle finding algorithm (e.g., Floyd) finds collision in $T \approx N^{0.5}$
  - Repeat about N times (randomizing f) to obtain N collisions in $T \approx N^{1.5}$

- Is tradeoff $T^2 \cdot S \approx C^2 \cdot N$ for collision search **optimal**?

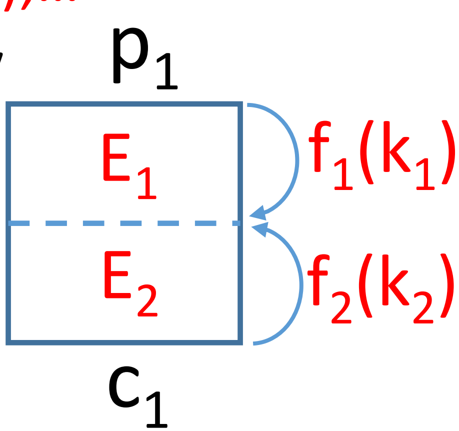


The Collision Pair Search Problem

- Is $T^2 \cdot S \approx C^2 \cdot N$ optimal?
- **Motivation**: breaking **double-encryption**
  - Assume $p, c, k_1, k_2 \in [N]$
  - Setting: given $(p_1,c_1),(p_2,c_2),\ldots$ find k_1,k_2



- **Best attack**: MITM gives $T \approx N$, but requires $S \approx N$
- Assume $S \approx 1$:
  - define $f_1(k_1)=E_1(p_1,k_1)$, $f_2(k_2)=(E_2)^{-1}(c_1,k_2)$
  - Find **collisions** $f_1(k_1)=f_2(k_2)$
  - **Test** each colliding candidate pair k_1,k_2 on $(p_2,c_2),\ldots$
- **Analysis**: each candidate k_1,k_2 **equally likely** to be correct
  - Need to find almost all $\approx N$ collision
  - Collision pair search problem with $C \approx N >> S \approx 1$
  - PCS gives $T^2 \approx C^2 \cdot N$ → with $C = N$ gives $T \approx N^{1.5}$



The Collision Pair Search Problem

- Is $T^2 \cdot S \approx C^2 \cdot N$ optimal?
- **Motivation**: if not optimal, can improve best-known time-space tradeoff for breaking **double-encryption**

- **Additional applications:** if not optimal, can improve best known time-space tradeoffs for various **MITM-type attacks** (in some parameter ranges):
  - Breaking **triple** (and **multiple**) **encryption**
  - Some **dedicated MITM attacks** on specific cryptosystems
  - Solving the **generalized birthday** problem
  - Solving the **subset-sum** problem
  - …

Our Results

- 1) Best-known time-space tradeoff $T^2 \cdot S \approx C^2 \cdot N$ for collision pair search problem is **optimal**
  - (for all parameters, in particular $S << C$)

- **Conclusion**: tradeoff algorithms for **applications cannot be improved** via more efficient collision search
- Can tradeoff algorithms for applications be improved **by other means**?
  - Unfortunately, **unconditional optimality proof** would overcome (variant of) **long-standing barrier** in complexity theory

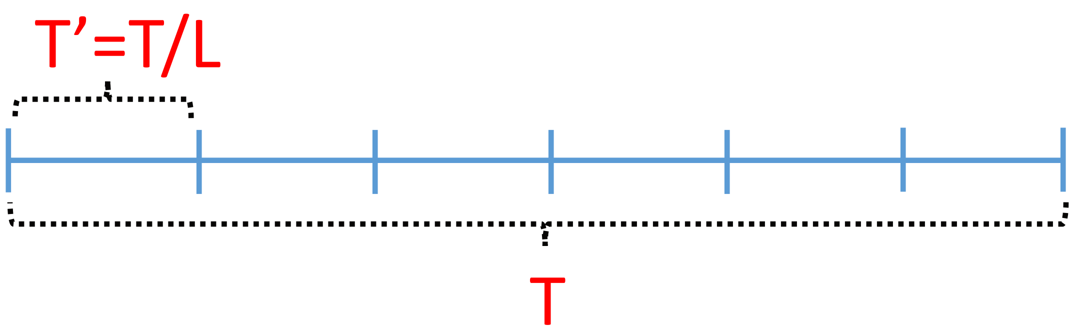
- 2) For breaking **double encryption**, we show that under **restriction**, best-known tradeoff is **optimal**

1st Result: Time-Space Tradeoff Lower Bounds for Collision Pair Search

- **Main idea** for proving optimality of $T^2 \cdot S \approx C^2 \cdot N$ of tradeoff:
- Adapt **framework** of Borodin and Cook ('82)
  - Based on the **branching program** model of computation
  - Previously used to derive several time-space tradeoff lower bounds (e.g., on sorting, matrix multiplication, FFT…)
  - Adaptation to collision search: first use in cryptography

Lower Bounds for Collision Pair Search: Proof Intuition

- 1) Divide T into L **time intervals** (of length T'=T/L)
  - Say algorithm **makes progress** in interval if it **outputs** C'=C/L collisions in interval
  - Consider **"mini-problem"**: output C' collisions in time T'
  - Prove: any "mini-algorithm" succeeds with **tiny** probability ≤ ε (over choice of f) – **independently of memory**
- 2) To output C collisions, algorithm outputs C'=C/L collisions in **some** interval
  - Some "mini-algorithm" (defined from **initial memory state of an interval**) must output C' collisions
  - By **union bound** over all $\leq 2^S$ "mini-algorithms", main alg succeeds w.p $\leq 2^S \cdot \varepsilon$
  - Need $\varepsilon \ll 2^{-S}$ to finish



Are Tradeoffs for Collision Search Applications optimal?

- **Cannot use framework** for proving optimality of collision search to prove optimality of applications
- In collision search: output length C is **long**
- In applications (e.g., breaking double encryption): output length is **short**
  - **Not clear** how to **measure progress** of algorithm towards solving problem

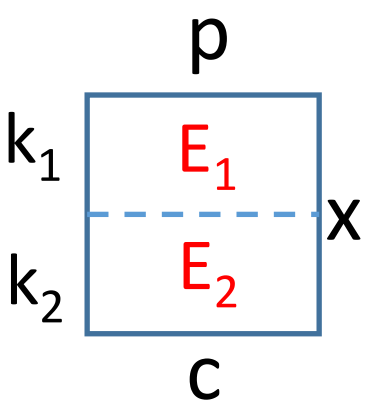
- **Long standing barrier** in complexity theory:
- Prove "meaningful" time-space tradeoff lower bound for **short-output problem** in **general computational model**
  - In **restricted** computational models (streaming, pebbling…), **strong lower bounds** are known

2^{nd} Result:
Time-Space Tradeoff Lower Bounds for Breaking Double Encryption

- Best known (PCS-based) time-space tradeoff $T^2 \cdot S \approx N^3$
- **Previous analysis**: Tessaro and Thiruvengadam (TCC'18) showed problem is **equivalent** to well-known **element-distinctness** (ED) problem
- Can we obtain **additional insight** into the problem?

Time-Space Tradeoff Lower Bounds for Breaking Double Encryption

- Is best known (PCS-based) time-space tradeoff $T^2 \cdot S \approx N^3$ optimal?
- Proving **unconditional** lower bound **very unlikely**



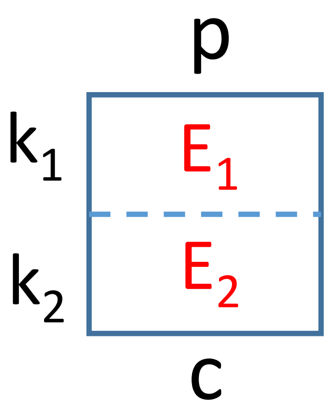
- Define new **restricted** computational model: **post-filtering model**

Post-Filtering Model

- **Post-filtering model**:
- Algorithm gets full access to a **part** of the input
- Access to remaining part **restricted** via **a post-filtering oracle**
  - Given 1st part of input, many **equally-likely** potential solutions exist
  - Algorithm **forced** to **produce many potential outputs** to be post-filtered by oracle

- Model **forces reduction** from **short-output** problem to related **long-output** problem

Post-Filtering Model for Breaking Double Encryption

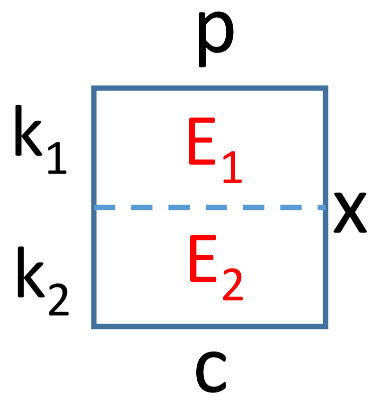


- Recall: best known attack only uses (p_2,c_2),… for **post-filtering** (k_1,k_2) candidates

- **In post-filtering model for double encryption** algorithm gets:
  - 1) Access to block cipher
  - 2) (p_1,c_1)
  - 3) Access to post-filtering oracle $O(k_1,k_2)$: return 1 for **correct key**
  - Can **only** be invoked on k_1,k_2 that encrypt p_1 to c_1

- **Captures** PCS-based attack and various **generalizations**

Post-Filtering Model for Breaking Double Encryption



- Algorithm gets:
  - 1) Access to block cipher
  - 2) (p_1,c_1)
  - 3) Access to post-filtering oracle $O(k_1,k_2)$: return 1 for **correct key**
  - Can **only** be invoked on k_1,k_2 that encrypt p_1 to c_1
- We prove tradeoff $T^2 \cdot S \approx N^3$ is **optimal** for **any post-filtering attack** on double encryption
  - Clean model abstracts away lower-level collision search problem
- **Conclusion**: to improve tradeoff, must non-trivially combine information form multiple (p_i,c_i)

Conclusions and Future Work

- Showed that best-known time-space tradeoff $T^2 \cdot S \approx C^2 \cdot N$ for collision pair search problem is **optimal**
- Presented the **post-filtering model** – a new restricted computational model
- For breaking double encryption: proved tradeoff $T^2 \cdot S \approx N^3$ optimal for **any post-filtering attack**

- Future work:
  - Extend **post-filtering** model to prove time-space lower bounds on additional problems
  - Alternatively, **bypass** the model and improve algorithms

Thanks for your attention!