Attacking Data Independent Memory Hard Functions

Jeremiah Blocki (Microsoft Research)
Joel Alwen (IST Austria)

ABSTRACT

We demonstrate an algorithm for evaluating data-independent memory-hard functions (iMHFs) with significantly less cumulative resources (e.g., memory/energy) than ideally desired of such algorithms. In particular we get that:

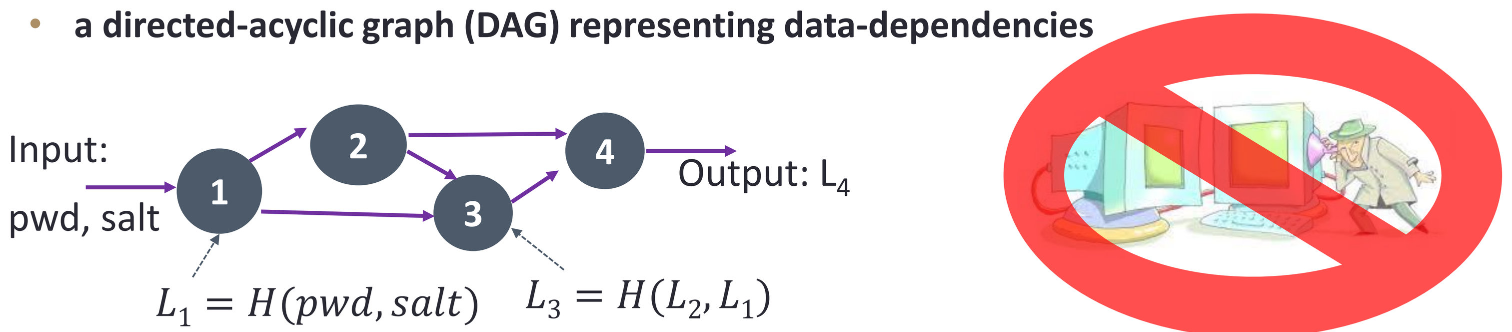
- Catena-Dragonfly and Catena-Butterfly can be computed by an algorithm with cumulative cost $O(n^{5/3})$ --- an improvement of $O(n^{1/3})$.
- Argon2i (winner of the Password Hashing Competition) can be computed by an algorithm with cumulative cost $\tilde{O}(n^{7/4})$ --- an improvement of $\tilde{O}(n^{1/4})$.
- *Any* iMHF can be computed by algorithm with cumulative cost $O\left(\frac{n^2}{log^{1-\varepsilon} n}\right)$ for any constant $\varepsilon > 0$--- an improvement of $O(log^{1-\varepsilon} n)$.

In particular, this shows that the goal of constructing an iMHF requiring $\Omega(n^2)$ cumulative resources is infeasible.

iMHF (Password Hash Function)

A data-independent memory hard function (iMHF) is defined by

- an underlying compression function H, and
- a directed-acyclic graph (DAG) representing data-dependencies



Advantage: Data-dependent MHFs (e.g., SCRYPT) are vulnerable to side-channel attacks due to their data-dependent memory access pattern.

Computing an iMHF (Pebbling)

Pebbling Rules:

- May place a pebble on node v_1 during any round.
- May remove a pebble from DAG in any round.
- May place a pebble on an unpebbled node v_i during round j only if all parents had pebbles on round j-1.

Pebbling Costs:

- (Each Round) pay energy cost (1 mwt) for each pebble --- cost to store value in memory.
- Pay energy cost $\bar{R}$ to place a new pebble on the DAG (e.g., $\bar{R} \approx 3{,}000$ mwt is cost to compute H)

Cumulative Cost of Pebbling Algorithm A:

$$cc(A) = \bar{R} \times (\#queries(H)) + \sum_{j=1}^{\#rounds} (\#pebbles(j))$$

Naïve Pebbling Algorithm *N*:

- Pebble graph in topological order (n rounds).
- Cumulative Cost:

$$cc(N) = O(n^2)$$

Attack Quality and Ideal iMHFs

(Amortized) Quality of Attack A

$$\text{Quality}(A) = \frac{CC(Naive)}{CC(A) \times \#inst(A)}$$

Amortized by #instances of iMHF computed.

c-Ideal iMHF

- For all attacks A, $\text{Quality}(A) \le c$
- DAG has constant indegree

Thm (Bad News): No c-Ideal iMHF exists for $c = O(log^{1-\varepsilon} n)$.

Significance?

- Cost of computing H varies greatly across architectures.
- **Contrast:** memory costs are consistent across architectures.

Depth-Robust DAGs are Necessary

Definition: We say that a DAG G=(V,E) is (e,d)-node robust if
$\forall S \subseteq V: \quad |S| \le e \Rightarrow \text{depth}(G - S) \ge d.$

Length of longest remaining path after removing nodes in S.

Theorem (Depth-Robustness is a necessary condition): If G is not (e,d)-node robust then is an (efficient) attack A such that

$$\text{Quality}(A) = \Omega\left(\max_{d \le g \le n}\left\{\frac{gn}{dn + g^2 + ge}\right\}\right).$$

Theorem (No DAG is sufficiently depth robust): If a DAG G=(V,E) has constant indegree then we can (efficiently) find $S \subseteq V$, s.t

$$|S| \le O(n/log^{1-\varepsilon} n) \text{ and } \text{depth}(G - S) \le {}^{n}/_{\log^2 n}$$

Note: yields attack with $\text{Quality}(A) = \Omega(log^{1-\varepsilon} n)$.

MAIN ATTACK (GEN-PEBBLE)

Algorithm 1: GenPeb (G, S, g, d)

```
Arguments: G = (V, E), S ⊆ V, g ∈ [depth(G − S), n], d ≥ depth(G − S)
for i = 1 to n do
    Pebble node i.
    l ← ⌊i/g⌋ * g + d + 1
    if i mod g ∈ [d] then                                   // Balloon Phase
        d' ← d − (i mod g) + 1
        N ← need(l, l + g, d')
        Pebble every v ∈ N which has all parents pebbled.
        Remove pebble from any v ∉ K where K ← S ∪ keep(i, i + g) ∪ {n}.
    else                                                    // Light Phase
        K ← S ∪ parents(i, i + g) ∪ {n}
        Remove pebbles from all v ∉ K.
    end
end
```

Light Phase: Discard most pebbles!

- Only keep pebbles on parents of next g nodes.
- One Light Phase: Cost = O(g|S|)
- All Light Phases: Total Cost = O(g|S|(n/g))= O(n|S|)

In ≤ d rounds we can recover all of the pebbles.

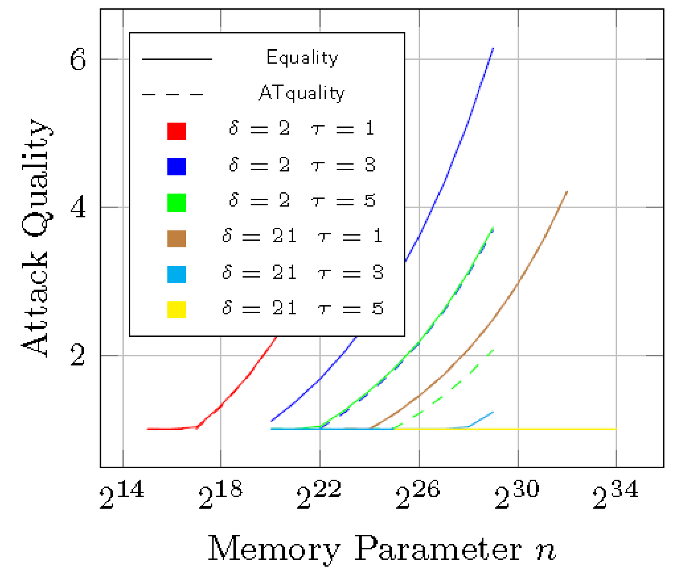
One Balloon Phase: Cost = O(dn)

All Balloon Phases: Total Cost= O(dn²/g).

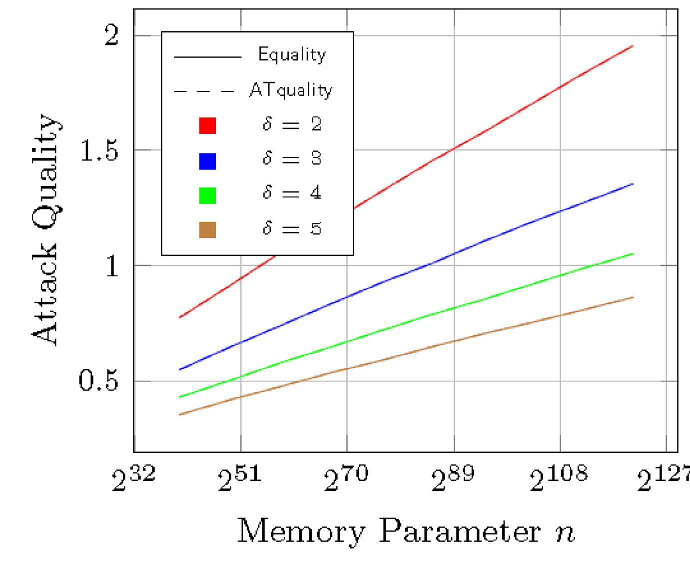
Each round of a balloon phase is potentially very expensive.

Key: balloon phase ends quickly!

PRACTICAL RESULTS



(a) Argon2i and SB

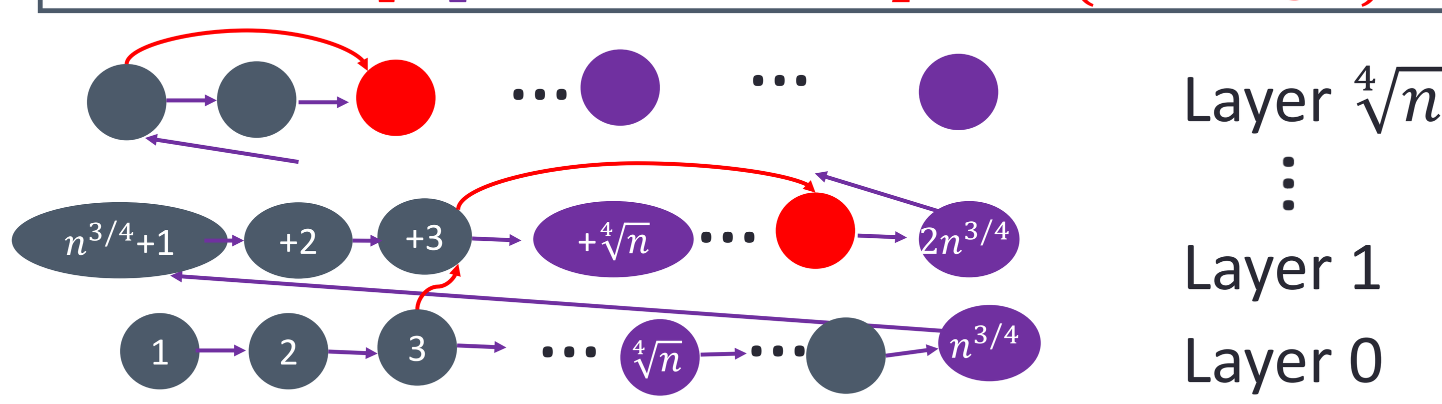


(b) Ideal iMHF

Attacking Argon2i

Argon2i DAG: G=(V,E), indegree=2
Edges: (v_i, v_{i+1}), $(v_{r(i)}, v_i)$ for $i \le n$ $\quad$ r(i)~Uniform([i-1])

Lemma: Let $S_1 = \{v_i | i = j \times \sqrt[4]{n}\}$, and $S_2 = \{v_i | v_{r(i)} \text{ and } v_i \text{ in same layer}\}$. Then depth($G - S_1 - S_2$) $\le \sqrt{n}$, and $E[S_2] = O(n^{3/4} \log n)$



CONCLUSION

Practical attacks against every (known) iMHFs:

- Argon2i
- Catena
- Balloon Hashing (New)

REFERENCES

- Password Hashing Competition (https://password-hashing.net/)
- Alex Biryukov ,Daniel Dinu and Dmitry Khovratovich. Argon2: new generation of memory-hard functions for password hashing and other applications. EURO S&P 2016.
- Christian Forler, Stefan Lucks and Jakob Wenzel. *Catena: A memory-consuming password scrambler*. ASIACRYPT 2014.
- Alex Biryukov and Dmitry Khovratovich. *Fast and Tradeoff-resilient memory-hard functions for cryptocurrencies and password hashing*. ASIACRYPT 2015.
- Cythia Dwork, Moni Naor and Hoeteck Wee. *Pebbling and proofs of work*. CRYPTO 2005.
- Joel Alwen and Vladimir Servinenko. *High Parallel Complexity Graphs an Memory-Hard Functions*. STOC 2015.
- Joel Alwen and Jeremiah Blocki. *Efficiently Computing Data-Independent Memory-Hard Functions*. http://eprint.iacr.org/2016/115