Introduction to Cryptology

7.3 - Hash functions: Generic Attacks

Federico Pintore

Mathematical Institute, University of Oxford (UK)

Michaelmas term 2020

The Birthday Problem

Suppose there are q people in a room.

The Birthday Problem

Suppose there are q people in a room.

- What is the probability that two people have the same birthday?

The Birthday Problem

Suppose there are q people in a room.

- What is the probability that two people have the same birthday?
- How many people are required to have a probability larger than $1/2$?

The Birthday Problem

Suppose there are q people in a room.

- What is the probability that two people have the same birthday?
- How many people are required to have a probability larger than 1/2 ?
  - The answer is 23:

$$\Pr(\text{all distinct}) = \frac{364}{365} \cdot \frac{363}{365} \cdot \ldots \cdot \frac{365-22}{365} < \frac{1}{2}$$

The Birthday Problem

Suppose you choose q elements uniformly in a set of N elements.

- What is the probability that two elements are equal?
- How large should q be with respect to N to have a probability larger than $1/2$?

The Birthday Problem

Suppose you choose q elements uniformly in a set of N elements.

- What is the probability that two elements are equal?
- How large should q be with respect to N to have a probability larger than $1/2$?

A formal solution to these problems in the following slides.

The Birthday Problem

Assume q balls are thrown to N bins ($q < \sqrt{2N}$), and denote by Coll the event that two balls end up being in the same bin.

The Birthday Problem

Assume q balls are thrown to N bins ($q < \sqrt{2N}$), and denote by Coll the event that two balls end up being in the same bin.

We can show that

$$q(q-1)/4N \leq \Pr(\text{Coll}) \leq q(q-1)/2N.$$

The Birthday Problem

Assume q balls are thrown to N bins ($q < \sqrt{2N}$), and denote by Coll the event that two balls end up being in the same bin.

We can show that

$$q(q-1)/4N \leq \Pr(\text{Coll}) \leq q(q-1)/2N.$$

Upper bound: For the event Coll_i (the i-th ball falls into an already occupied bin), it holds that $\Pr(\text{Coll}_i) \leq (i-1)/N$ (at most $i-1$ bins are already occupied).

$$\Pr(\text{Coll}) = \Pr(\bigcup_{i=1}^{q} \text{Coll}_i) \leq \sum_{i=1}^{q} \Pr(\text{Coll}_i) \leq$$

$$\leq 0/N + \cdots + (q-1)/N = \frac{q(q-1)}{2N}$$

The Birthday Problem

Lower bound: For the event NoColl_i (no collisions after throwing the i-th ball), it holds

$$\Pr(\text{NoColl}_{i+1}|\text{NoColl}_i) = (N - (i))/N. \tag{1}$$

We note

$$\Pr(\text{NoColl}_q) = 1 - \Pr(\text{Coll}). \tag{2}$$

Now,

$$\Pr(\text{NoColl}_q) = \Pr(\text{NoColl}_q \cap \text{NoColl}_{q-1}) =$$

$$= \Pr(\text{NoColl}_q|\text{NoColl}_{q-1}) \cdot \Pr(\text{NoColl}_{q-1})$$

and, iterating the above reasoning, we obtain:

$$\Pr(\text{NoColl}_q) = \prod_{i=1}^{q-1} \Pr(\text{NoColl}_{i+1}|\text{NoColl}_i). \tag{3}$$

The Birthday Problem

From equations (1), and (3) we obtain

$$\Pr(\text{NoColl}_q) = \prod_{i=1}^{q-1} \left(1 - \frac{i}{N}\right) \tag{4}$$

Since $1 - x \leq e^{-x}$ when $x \leq 1$ and $i/N < 1$, it holds

$$\Pr(\text{NoColl}_q) \leq e^{-\sum_{i=1}^{q-1}(i/N)} = e^{-q(q-1)/2N}. \tag{5}$$

Thanks to equation (2)

$$\Pr(\text{Coll}) \geq 1 - e^{-q(q-1)/2N}$$

where

$$1 - e^{-q(q-1)/2N} \geq q(q-1)/4N$$

since $q < \sqrt{2N}$ and $e^{-x} \leq 1 - x/2$ when $|x| \leq 1$.

The Birthday Problem

The probability of having a collision is $\approx 1/2$ when $q \approx N^{1/2}$.

The Birthday Problem

The probability of having a collision is $\approx 1/2$ when $q \approx N^{1/2}$.

How does the birthday problem relate to hash functions?

Hash functions: The Birthday attack

Consider the function $F : \{0,1\}^* \to \{0,1\}^\ell$. Its range has size $N = 2^\ell$.

When $q \approx 2^{\ell/2}$ evaluations are executed, the probability of finding a collision is $\approx 1/2$.

Hash functions: The Birthday attack

Consider the function $F : \{0,1\}^* \to \{0,1\}^\ell$. Its range has size $N = 2^\ell$.

When $q \approx 2^{\ell/2}$ evaluations are executed, the probability of finding a collision is $\approx 1/2$.

There are no generic attacks for preimage and second-preimage resistance.

Hash Functions: A Better Birthday Attack

The original birthday attack uses lots of memory storage, as it has to store $\mathcal{O}(q) = \mathcal{O}\left(2^{\ell/2}\right)$ digests.

Managing storage for 2^{60} bytes is often more difficult than executing 2^{60} CPU instructions.

Hash Functions: A Better Birthday Attack

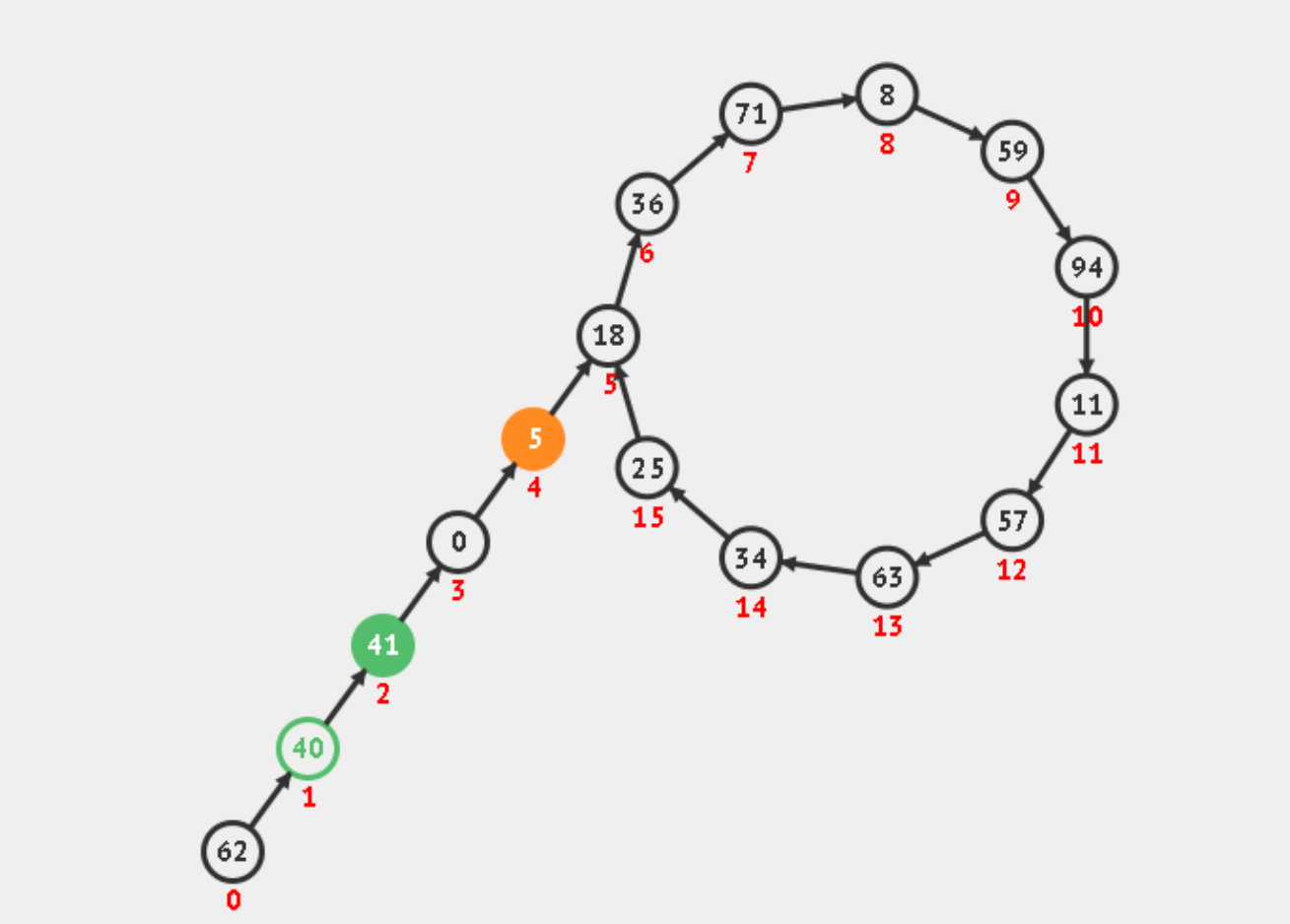
The original birthday attack uses lots of memory storage, as it has to store $\mathcal{O}(q) = \mathcal{O}\left(2^{\ell/2}\right)$ digests.

Managing storage for 2^{60} bytes is often more difficult than executing 2^{60} CPU instructions.

Anything better?

Hash Functions: A Better Birthday Attack

The Floyd's cycle finding idea can be exploited!



https://visualgo.net/bn/cyclefinding

Hash Functions: A Better Birthday Attack

Let x_0 be a uniform input of the H and denote $H^i(x_0)$ by x_i.

Suppose $x_I = x_J$ for some distinct $I, J \in [1, q]$.

Then, there is $i < J$ s.t. $x_i = x_{2i}$.

Hash Functions: A Better Birthday Attack

Let x_0 be a uniform input of the H and denote $H^i(x_0)$ by x_i.

Suppose $x_I = x_J$ for some distinct $I, J \in [1, q]$.

Then, there is $i < J$ s.t. $x_i = x_{2i}$.

- From x_I, the sequence is periodic with a period Δ that divides $J - I$;
- let $i < J$ be the smallest multiple of Δ greater than or equal to I ($x_I, x_{I+1}, \ldots, x_{J-1}$ contains Δ consecutive indices);
- $x_i = x_{2i}$ since $i > I$ and $2i - i = i$ is a multiple of Δ.

Hash Functions: A Better Birthday Attack

Let x_0 be a uniform input of the H and denote $H^i(x_0)$ by x_i.

Suppose $x_I = x_J$ for some distinct $I, J \in [1, q]$.

Then, there is $i < J$ s.t. $x_i = x_{2i}$.

- From x_I, the sequence is periodic with a period Δ that divides $J - I$;
- let $i < J$ be the smallest multiple of Δ greater than or equal to I ($x_I, x_{I+1}, \ldots, x_{J-1}$ contains Δ consecutive indices);
- $x_i = x_{2i}$ since $i > I$ and $2i - i = i$ is a multiple of Δ.

The index i is found after an exhaustive search (in the i-th iteration $H(x_{i-1})$ and $H(H(x_{2(i-1)}))$ are computed).

Hash Functions: A Better Birthday Attack

Starting from x_0 and after a maximum of i-steps, a cycle is entered.

The goal is finding the *first* element of the cycle. To this end, two starting points, $x := x_0$ and $x' := x_i$, are considered.

- At each step, x and x' are updated with an evaluation of H;
- while x is out of the cycle, $x \neq x'$, since x' is moving within the cycle;
- as soon x enters the cycle, it *meets* x', i.e. $x = x'$. Indeed, if j is the smallest index for which x_j is in the cycle, then $x' = x_{i+j}$ where $i \equiv 0 \pmod{\Delta}$.

Hash Functions: A Better Birthday Attack

Starting from x_0 and after a maximum of i-steps, a cycle is entered.

The goal is finding the *first* element of the cycle. To this end, two starting points, $x := x_0$ and $x' := x_i$, are considered.

- At each step, x and x' are updated with an evaluation of H;
- while x is out of the cycle, $x \neq x'$, since x' is moving within the cycle;
- as soon x enters the cycle, it *meets* x', i.e. $x = x'$. Indeed, if j is the smallest index for which x_j is in the cycle, then $x' = x_{i+j}$ where $i \equiv 0 \pmod{\Delta}$.

Also j is found after an exhaustive search.

We note that x_{j-1} and x_{i+j-1} form a collision (one point out of the cycle, the other inside).

A Better Birthday Attack

Let $H : \{0,1\}^* \to \{0,1\}^\ell$ be an unkeyed hash function. The goal is finding x, x' s.t. $H(x) = H(x')$.

$x_0 \leftarrow\!\!\$\ \{0,1\}^{\ell+1}$

$x' := x_0, x := x_0, \ell := 0$

for $i = 1, 2, \cdots$ do

  $\ell = \ell + 1$

  $x = H(x) = H^{(i)}(x_0)$

  $x' = H(H(x')) = H^{(2i)}(x_0)$

  if $x = x'$ break

$x' := x, x := x_0, i := \ell$

for $j = 1 \cdots, i$

  if $H(x) = H(x')$ return x, x'

  else $x = H(x) = H^{(j)}(x_0)$

    $x = H(x') = H^{(i+j)}(x_0)$

Hash Functions: A Better Birthday Attack

The algorithm has approximately the same time complexity of the birthday attack. The success probability is the same.

It only requires $\mathcal{O}(1)$ memory, namely,
storage of two digests at each step!

Further Reading I


Mihir Bellare and Phillip Rogaway.
Random oracles are practical: A paradigm for designing efficient protocols.
In Proceedings of the 1st ACM conference on Computer and communications security, pages 62–73. ACM, 1993.


Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche.
Keccak sponge function family main document.
Submission to NIST (Round 2), 3:30, 2009.


Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya.
Merkle-Damgård revisited: How to construct a hash function.
In Advances in Cryptology–CRYPTO 2005, pages 430–448. Springer, 2005.

Further Reading II


Morris J Dworkin.
SHA-3 standard: Permutation-based hash and extendable-output function.
No. Federal Inf. Process. Stds.(NIST FIPS)-202, 2015.


Pierre Karpman, Thomas Peyrin, and Marc Stevens.
Practical free-start collision attacks on 76-step SHA-1.
In Advances in Cryptology–CRYPTO 2015, pages 623–642. Springer, 2015.


Neal Koblitz and Alfred J Menezes.
The random oracle model: a twenty-year retrospective.
Designs, Codes and Cryptography, pages 1–24, 2015.

Further Reading III

Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone.
Handbook of applied cryptography.
CRC press, 1996.

Marc Stevens.
New collision attacks on SHA-1 based on optimal joint local-collision analysis.
In Advances in Cryptology–EUROCRYPT 2013, pages 245–261. Springer, 2013.

Marc Stevens, Elie Bursztein, Pierre Karpman, Ange Albertini, and Yarik Markov.
The first collision for full SHA-1.
In Annual International Cryptology Conference–CRYPTO 2017, pages 570–596. Springer, CHam, 2005.