Hash Functions

Benjamin Pecson

11/20/2019

Cryptographic Hash Functions should have the following three properties:

- Preimage Resistance
- Weak Collision Resistance
- Strong Image Resistance

The Discrete Logarithm Hash has strong image resistance.

The Discrete Logarithm Hash is comprised of:

- Two primes, p and q
  Where, p=2q+1
- Two different Primitive Roots:
  Alpha and Beta, where:
  - $\alpha^a \equiv \beta \pmod{p}$
  - $\beta^b \equiv \alpha \pmod{p}$

Finidng a and b in the equations above is difficult.

Hash a message m which is less than q^2
$h(m) < p$

The input produces a digest such that $m < p$

Hash Function: $h(m) = h(x_1 + x_2 * q) \equiv \alpha^{x_1}\beta^{x_2}$(mod p)

$m < q^2$, so write m in base q

Where $m = x_1 + x_2 * q$
and $0 \leq x_1,\ x_2 < q$

How to prove this hash function is preimage resistant:

If we can find a collision to this discrete log hash then we can use the discrete log problem $\alpha^a \equiv \beta$ (mod p)

Suppose a collision is found:

$m = x_1 + x_2 * q$,
where $m! = m$‘,
but m‘ $= y_1 + x_2 * q$,
and $h(m) = h(m$‘$)$

$\alpha^{x_1}\beta^{x_2} \equiv \alpha^{y_1}\beta^{y_2}$ (mod p)
$\alpha^{x_1}(\alpha^a)^{x_2} \equiv \alpha^{y_1}(\alpha^a)^{y_2}$ (mod p)
$\alpha^{x_1+ax_2} \equiv \alpha^{y_1+ay_2}$ (mod p)
$\alpha^{(x_1-y_1+a(x_2-y_2))} \equiv 1$ (mod p)

Because p is a primitive root,

$p - 1|(x_1 - y_1 + a(x_2 - y_2)$

$(x_1 - y_1 + a(x_2 - y_2) \equiv 0$(mod p-1)

$a(x_2 - y_2) \equiv -(x_1 - y_1)(y_1 - x_1)$(mod p-1)

$a \equiv (x_2 - y_2)^{-1}(y_1 - x_1)$(mod p-1)

Why is this not used in practice?

Because, even with small input there is a lot of calculations.
Hash Attacks

Birthday Attack

How many people should exist in a room such that the probability two people share the same birthday is approx 1/2?

There are 365 days in a year, and there are two people

Probability they share a birthday? 1/365

Probability that they don't share the same birthday? 1-1/365

What if there is three people?
$(1 - 1/365)(1 - 2/365)$
What if there is k people?
$(1 - 1/365)(1 - 2/365)...(1 - (k - 1)/365) \approx e^{-k^2/2(365)}$

Find k such that
$1/2 \leq e^{-k^2/2(365)}$, k = 23

Another Example

Take k things being chosen randomly from n possibilities (with replacement). Then the probability that no two choices are the same is $\approx e^{-k^2/2^n}$

License Plate Example

How many cars are needed before it is likely that two cars have the same last three digits?

N=1000 possible endings

The probability should be: $e^{-k^2/2^n} < 1/2$

$e^{-k^2/2000} < 1/2$
$= -k^2/1000 < ln(1/2)$
$= -ln(2) - k^2 < -2000ln(2)$
$-k^2 > ln(2)$
$k > (2000ln(2))^{-1/2} \approx 37$

How Birthday Attack works:

Alice has a digital signature function s(x)
and a hash function h(x)

Alice signs a message using s(h(x))** This is preimage resistant, no one else will have the same x

Eve's goal is to have Alice sign a bad contract.

Eve drafts two contracts: 1 good and 1 bad

Eve wants to find 35 places (35 is a reasonable example number for a hash example) she can make a small change without changing the meaning of the message.

2^{35} good contracts and 2^{35} bad contracts
2^{36} contracts total

Eve hashes all of them to find a collision. What is the probability of finding a solution?

Probability of number collisions: 2^{36}, $k = 2^{36}$

$e^{-k^2/2(n)}$, $n = 2^{60}$ digests, $k = 2^{36}$ contracts

$e^{-2^{72}/2^{61}} = e^{-2^{11}} = e^{-2048} \approx 10^{-800}$, which is a very small chance for collision

There will be a collision between a good contract and a bad contract.

Eve has two contracts
MG and MB
Eve gives MG to Alice, Alice signs it.

s(h(MG))

Alice has the valid signature, which is valid for both the good contract and the bad contract.

h(MG)=h(mb), same signature is valid for both

How does Alice protect against this?
Eve chooses the MG using a specific method
If Alice makesa single change the hash of MB will not equal the hash of MB
So, Alice ultimately produces the final contract.