Discrete logarithm problem IV

Pollard's rho method

Tanja Lange

Eindhoven University of Technology

2MMC10 – Cryptology

Pollard's rho method

The motivation is to remove the storage costs by turning solving the DLP into a random walk on elements of G.

This is a common strategy which we will also encounter for factorization and collision attacks for hash functions (cryptographic functions mapping to short outputs for which it should be hard to find collisions).

By the birthday paradox, after $\sqrt{\pi n/2}$ random draws from a set of n elements, one element will be drawn twice with 50% probability.

Two problems need to be solved to use the rho method successfully:

1. Design step function so that it "randomly" samples elements (so that the birthday paradox applies) while being deterministic (so we can use Floyd's cycle finding method to remove storage).
2. Design step function so that collision gives meaningful result.

Pollard's rho method

The motivation is to remove the storage costs by turning solving the DLP into a random walk on elements of G.

This is a common strategy which we will also encounter for factorization and collision attacks for hash functions (cryptographic functions mapping to short outputs for which it should be hard to find collisions).

By the birthday paradox, after $\sqrt{\pi n/2}$ random draws from a set of n elements, one element will be drawn twice with 50% probability.

Two problems need to be solved to use the rho method successfully:

1. Design step function so that it "randomly" samples elements (so that the birthday paradox applies) while being deterministic (so we can use Floyd's cycle finding method to remove storage).
2. Design step function so that collision gives meaningful result.

For DLP want $W_i = a_i P + b_i Q$ so that $W_i = W_j$ means that

$$a_i P + b_i Q = a_j P + b_j Q \Leftrightarrow (b_i - b_j)Q = (a_j - a_i)P \Leftrightarrow a \equiv (a_j - a_i)/(b_i - b_j) \bmod n.$$

This means that collisions are meaningful if $\gcd(n, b_i - b_j) = 1$.

Pollard rho for DLP: Attempt 1

For DLP want $W_i = a_i P + b_i Q$ so that $W_i = W_j$ means that

$$a_i P + b_i Q = a_j P + b_j Q \Leftrightarrow (b_i - b_j) Q = (a_j - a_i) P \Leftrightarrow a \equiv (a_j - a_i)/(b_i - b_j) \bmod n.$$

Simplest approach: let $g, h : G \to [0, n-1]$ and

$$f(W) = g(W)P + h(W)Q.$$

Then $a_i = g(W_{i-1}), b_i = h(W_{i-1})$.

If g and h are sufficiently different and random,
f provides a random walk on G.

Pollard rho for DLP: Attempt 1

For DLP want $W_i = a_i P + b_i Q$ so that $W_i = W_j$ means that

$$a_i P + b_i Q = a_j P + b_j Q \Leftrightarrow (b_i - b_j)Q = (a_j - a_i)P \Leftrightarrow a \equiv (a_j - a_i)/(b_i - b_j) \bmod n.$$

Simplest approach: let $g, h : G \to [0, n-1]$ and

$$f(W) = g(W)P + h(W)Q.$$

Then $a_i = g(W_{i-1}), b_i = h(W_{i-1})$.

If g and h are sufficiently different and random,
f provides a random walk on G.

Pick a random starting point

$$W_0 = a_0 P + b_0 Q.$$

Each step costs one double-scalar multiplication.
Total cost: $\sqrt{\pi n/2}$ double-scalar multiplications.

Pollard rho for DLP: Attempt 1

For DLP want $W_i = a_i P + b_i Q$ so that $W_i = W_j$ means that

$$a_i P + b_i Q = a_j P + b_j Q \Leftrightarrow (b_i - b_j)Q = (a_j - a_i)P \Leftrightarrow a \equiv (a_j - a_i)/(b_i - b_j) \bmod n.$$

Simplest approach: let $g, h : G \to [0, n-1]$ and

$$f(W) = g(W)P + h(W)Q.$$

Then $a_i = g(W_{i-1}), b_i = h(W_{i-1})$.

If g and h are sufficiently different and random,
f provides a random walk on G.

Pick a random starting point

$$W_0 = a_0 P + b_0 Q.$$

Each step costs one double-scalar multiplication.
Total cost: $\sqrt{\pi n/2}$ double-scalar multiplications.

A double-scalar multiplication is cheaper than two separate scalar multiplications. (Share doublings, add P, Q, or $P + Q$.)

Pollard rho for DLP: Attempt 2 – additive walks

[Note: We have a solution, we're just haggling about the price.]

Idea: we do not need n^2 directions to look random.
Having some fixed set of step directions suffices.

Additive walks make each step cheaper, try to keep features.

Pollard rho for DLP: Attempt 2 – additive walks

[Note: We have a solution, we're just haggling about the price.]

Idea: we do not need n^2 directions to look random.
Having some fixed set of step directions suffices.

Additive walks make each step cheaper, try to keep features.

Pick small number, e.g. $k = 32$, of random $(c_j, d_j) \in [0, n-1]^2$.
Compute and store k steps $S_j = c_j P + d_j Q, 0 \leq j < k$.

Fixed (small) number of double-scalar multiplications,
fixed (small) amount of storage.

Define the step function, costing 1 ADD per step, as

$$f(W) = W + S_{s(W)}, \text{ with } s : G \to [0, k-1].$$

s assigns one of the precomputed k steps to each group element

Pollard rho for DLP: Attempt 2 – additive walks

[Note: We have a solution, we're just haggling about the price.]

Idea: we do not need n^2 directions to look random.
Having some fixed set of step directions suffices.

Additive walks make each step cheaper, try to keep features.

Pick small number, e.g. $k = 32$, of random $(c_j, d_j) \in [0, n-1]^2$.
Compute and store k steps $S_j = c_j P + d_j Q, 0 \leq j < k$.

Fixed (small) number of double-scalar multiplications,
fixed (small) amount of storage.

Define the step function, costing 1 ADD per step, as

$$f(W) = W + S_{s(W)}, \text{ with } s : G \to [0, k-1].$$

s assigns one of the precomputed k steps to each group element

Typical choice: take $s(W) \equiv x(W) \bmod k$, where $x(W)$ is the x-coordinate of W and $\mathbf{F}_p$ is represented as integers in $[0, p-1]$.
This is why we use affine not projective coordinates. Want unique representation.

Rho for DLP with additive walks

$$f(W) = W + S_{s(W)}, \text{ with } s : G \to [0, k-1].$$

Problem 1: we don't know anymore how to write $W = aP + bQ$.

Rho for DLP with additive walks

$$f(W) = W + S_{s(W)}, \text{ with } s : G \to [0, k-1].$$

Problem 1: we don't know anymore how to write $W = aP + bQ$.

Solution: start from *known* starting point and keep track of the scalars.

Each step updates

$$W \leftarrow W + S_{s(W)}, \quad a \leftarrow a + c_{s(W)}, \quad b \leftarrow b + d_{s(W)}.$$

starting from $W = a_0 P + b_0 Q, a = a_0, b = b_0$. (Known random a_0, b_0.)

Rho for DLP with additive walks

$$f(W) = W + S_{s(W)}, \text{ with } s : G \to [0, k-1].$$

Problem 1: we don't know anymore how to write $W = aP + bQ$.

Solution: start from *known* starting point and keep track of the scalars.

Each step updates

$$W \leftarrow W + S_{s(W)}, \quad a \leftarrow a + c_{s(W)}, \quad b \leftarrow b + d_{s(W)}.$$

starting from $W = a_0P + b_0Q, a = a_0, b = b_0$. (Known random a_0, b_0.)

Problem 2: How big does k need to be for f to look random?

Rho for DLP with additive walks

$$f(W) = W + S_{s(W)}, \text{ with } s : G \to [0, k-1].$$

Problem 1: we don't know anymore how to write $W = aP + bQ$.

Solution: start from *known* starting point and keep track of the scalars.

Each step updates

$$W \leftarrow W + S_{s(W)}, \quad a \leftarrow a + c_{s(W)}, \quad b \leftarrow b + d_{s(W)}.$$

starting from $W = a_0 P + b_0 Q, a = a_0, b = b_0$. (Known random a_0, b_0.)

Problem 2: How big does k need to be for f to look random?

Additive walks induce anti-collisions (see next page), this delays collisions.
If there are k steps then the runtime increases by a factor of

$$1/\sqrt{1 - 1/k} \approx 1 + 1/(2k).$$

Anti-collisions in additive walks

Fix point T. Let W and W' be two independent uniform random points.

Anti-collisions in additive walks

Fix point T. Let W and W' be two independent uniform random points.

$W \neq W'$ both map to T if simultaneously for $i \neq j$:

$$T = W + S_i = W' + S_j, \quad s(W) = i, \quad s(W') = j.$$

These conditions have probability $1/n^2$, $1/k$, and $1/k$ respectively.

Anti-collisions in additive walks

Fix point T. Let W and W' be two independent uniform random points.

$W \neq W'$ both map to T if simultaneously for $i \neq j$:

$$T = W + S_i = W' + S_j, \quad s(W) = i, \quad s(W') = j.$$

These conditions have probability $1/n^2$, $1/k$, and $1/k$ respectively.

Summing over all (i, j) gives the overall probability

$$\sum_{i \neq j} 1/(kn)^2 = k(k-1)/(kn)^2 = (1 - 1/k)/n^2.$$

Anti-collisions in additive walks

Fix point T. Let W and W' be two independent uniform random points.

$W \neq W'$ both map to T if simultaneously for $i \neq j$:

$$T = W + S_i = W' + S_j, \quad s(W) = i, \quad s(W') = j.$$

These conditions have probability $1/n^2$, $1/k$, and $1/k$ respectively.

Summing over all (i, j) gives the overall probability

$$\sum_{i \neq j} 1/(kn)^2 = k(k-1)/(kn)^2 = (1 - 1/k)/n^2.$$

Collisions can occur at any T, so add over the n choices of T.
The probability of immediate collision from W and W' is

$$(1 - 1/k)/n$$

instead of $1/n$. Thus $\sqrt{\pi n/2}$ changes to $\sqrt{\pi n/(2(1 - 1/k))}$.

Anti-collisions in additive walks

Fix point T. Let W and W' be two independent uniform random points.

$W \neq W'$ both map to T if simultaneously for $i \neq j$:

$$T = W + S_i = W' + S_j, \quad s(W) = i, \quad s(W') = j.$$

These conditions have probability $1/n^2$, $1/k$, and $1/k$ respectively.

Summing over all (i, j) gives the overall probability

$$\sum_{i \neq j} 1/(kn)^2 = k(k-1)/(kn)^2 = (1 - 1/k)/n^2.$$

Collisions can occur at any T, so add over the n choices of T.
The probability of immediate collision from W and W' is

$$(1 - 1/k)/n$$

instead of $1/n$. Thus $\sqrt{\pi n/2}$ changes to $\sqrt{\pi n/(2(1 - 1/k))}$.

Additive walks need more steps by a factor of

$$1/\sqrt{1 - 1/k} \approx 1 + 1/(2k).$$

Schoolbook method for Pollard rho

The schoolbook method uses a step function with only 3 types of steps.

This would be very bad for randomness with the $1 + 1/(2k)$ formula!

The method escapes that by using doubling as one of the steps.

$$W \leftarrow \begin{cases} W+P \\ W+Q \\ 2W \end{cases}, a \leftarrow \begin{cases} a+1 \\ a \\ 2a \end{cases}, b \leftarrow \begin{cases} b \\ b+1 \\ 2b \end{cases}, \text{ for } s(W) = \begin{cases} 0 \\ 1 \\ 2 \end{cases}.$$

Typically $s(W)$ takes an integer representation of $x(W)$ and outputs the remainder of division by 3.