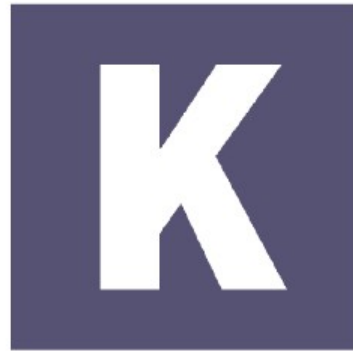


# Push



K r y p t o M a g i k

# Push

## Key Exchange Algorithm

# Push

- Invented in 2019
- Reference implementation in Python

# Design Goals

- To be resistant to DLP attacks
- To have smaller key lengths than traditional DiffieHellman

# Key Generation

- Generate 2  $N$  bit primes, let  $A$  be the secret modulus and let  $B$  be the public key
- Combine the product of  $A$  and  $B$  to produce the public modulus
- Choose a integer in  $B$  between 1 and  $N - 1$  and let that be the secret key

# Key Exchange Setup

- Alice and Bob generate SK, PK, N
- Either Bob or Alice sends their public key over
- Alice and Bob both send their public modulus and calculate the umbrella U by the product of the two modulus
- Alice and Bob calculate the shared modulus S by the product of their secret modulus and the other's public modulus
- Either Alice or Bob sends S

# Key Exchange Phase 1

- Alice and Bob have chosen to use Bob's public key. Alice and Bob both raise Bob's public key to their secret exponent modulo the umbrella modulus,  $U$ . They exchange phase 1.

# Key Exchange Phase 2

- Alice and Bob compute phase1 raised to the shared modulus  $S$  arriving at the secret key.



# Cryptanalysis

- TBD