Push

KryptoMagik

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 them be A and B and let them not be equal
- Let A be the private modulus M
- Combine the product of A and B to produce the public modulus N
- Choose a integer in N between 1 and N – 1 and let that be the secret key, SK

Key Exchange Setup

- Alice and Bob generate SK, N, M
- Alice and Bob send their public modulus N to each other and compute U as the produce of NA and NB
- They compute S as product of the private modulus M and U and exchange S values
- They then come to SS as the product of the two S values
- Alice and Bob both generate an ephemeral key Y between 1 and SS – 1 and exchange Y values

Key Exchange Phase 1

- Alice and Bob both raise Bob’s Y to their secret exponent modulo SS. They exchange phase 1.

Key Exchange Phase 2

- Alice and Bob compute phase1 raised to the shared modulus SS arriving at the secret modulus.

Key Exchange Phase 3

- Alice and Bob raise Alice’s Y to the power of their secret key modulo the secret modulus and exchange phase3

Key Exchange Phase 4

- Alice and Bob raise each other’s phase3 to their secret exponent and arrive at the shared key

Cryptanalysis

- TBD