Diffie-Hellman Key-Exchange Protocol
Agree on (public) GRP parameters $p, g$

[optionally: $q = lpf(p-1)$]

A

- Chooses Secret $x$
- Computes public $g^x \mod p$

B

- Chooses Secret $y$
- Computes public $g^y \mod p$

$g^x \mod p$

$g^y \mod p$

\[
\begin{align*}
[ g^y \mod p ]^x \mod p &= \text{key material} \\
g^{xy} \mod p &= g^x \mod p \cdot y \mod p \\
g^{xy} \mod p &= \text{shared key}
\end{align*}
\]

\[
\begin{align*}
[ g^x \mod p ]^y \mod p &= \text{key material} \\
g^{yx} \mod p &= g^y \mod p \cdot x \mod p \\
g^{yx} \mod p &= \text{shared key}
\end{align*}
\]

Shared key determination is based on the computational complexity of finding $x$ ($y$), given $g, p, g^x \mod p (g^y \mod p)$; i.e., of computing discrete logarithms.
Man-in-the-Middle Attack $\Rightarrow$ no Authentication

A chooses secret $x$ and computes $g^x \mod p$.

B chooses secret $y$ and computes $g^y \mod p$.

M chooses secret $m$ and computes $g^m \mod p$.

$[g^x \mod p]^m \mod p = g^{mx} \mod p = g^{xm} \mod p = K_{am}$

$[g^y \mod p]^m \mod p = g^{my} \mod p = g^{ym} \mod p = K_{bm}$

Problem 1: Key Exchange without Authentication
Problem 2: Reuse of $x, y \Rightarrow$ replay and forced reuse of shared key; timing attack
Potential Solutions
(not mutually exclusive)

1. Secure, published associations: \( A \leftarrow \rightarrow (g_A, p_A, g_A^x \mod p_A) \)
   
   \( \Rightarrow \) equivalent of using signed, public-key certificates

2. Establish secure dependency of key exchange on prior, independent authentication
   
   \( \Rightarrow \) use of other keys for mutual authentication

3. Establish private, shared groups \((g, p: q)\) between two communicating parties
   
   \( \Rightarrow \) use of independent protocols for group sharing, privacy
   
   \( (\text{separate multicast groups})\)

4. Use explicit replay-detection mechanisms; e.g., nonces (and PK encryption)

Note: Potential solutions depend on other security protocols
Discrete Logarithms (aka. indices)

1. Primitive roots of modulus p

- let g and p be relatively prime (note: p does not have to be a prime number)

- consider all m for which \( g^m \equiv 1 \mod p \)

  o minimum m is the order of g mod p,
  the length of period generated by g
  the exponent to which g belongs (modp)

  o maximum \( m = \phi(p) \), by Euler's theorem, where \( \phi(p) \) is the totient of p

- if g is of the order \( \phi(p) \) , then g is a primitive root of p, which means that:

  \[ g^1 \mod p, g^2 \mod p, \ldots, g^{\phi(p)} \mod p \]
  - are distinct and represent a permutation of \{1, ..., p-1\}
  - are relatively prime to p
  - if p is prime, \( \phi(p) = p-1 \); so the set size (length of period) is p-1

Note: the only integers with primitive roots are those of the form 2, 4, \( p^a \), \( 2p^a \) where p is any (odd) prime
Discrete Logarithms (aka. indices) -ctnd

2. Properties of Discrete Logarithms

Observation
- any integer \( x = r \mod p \) for any \( r , p \) where \( 0 \leq r \leq p-1 \)
- if \( g \) is a primitive root of prime \( p \), \( x = g^i \mod p \), where \( 0 \leq i \leq p-1 \)

Definition
- exponent \( i \) is the index (discrete log) of \( x \) in base \( g \mod p \); i.e., \( \text{ind}_{g,p}(x) \)

Ordinary Logarithms

1. Definition : \( x = b \log_b(x) \)
2. \( \log_b(1) = 0 \)
3. \( \log_b(b) = 1 \)
4. \( \log_b(ab) = \log_b(a) + \log_b(b) \)
4a. \( \log_b(a^r) = r \times \log_b(a) \)

Discrete Logarithms

1. Definition : \( x = g \text{ind}_{g,p}(x) \)
2. \( \text{ind}_{g,p}(1) = 0 \)
3. \( \text{ind}_{g,p}(g) = 1 \)
4. \( \text{ind}_{g,p}(xy) = [ \text{ind}_{g,p}(x) + \text{ind}_{g,p}(y) ] \mod \phi(p) \)
4a. \( \text{ind}_{g,p}(x^r) = r \times [ \text{ind}_{g,p}(x) ] \mod \phi(p) \)

* Proof: \( g^{\text{ind}_{g,p}(xy)} \mod p = (g^{\text{ind}_{g,p}(x)} \mod p)(g^{\text{ind}_{g,p}(y)} \mod p)(g^{k \phi(p)} \mod p) \)

\[ = 1 \]

\[ = [ g^{\text{ind}_{g,p}(x)} + \text{ind}_{g,p}(y) + k \phi(p) ] \mod p \]

Hence, \( \text{ind}_{g,p}(xy) = [ \text{ind}_{g,p}(x) + \text{ind}_{g,p}(y) ] \mod \phi(p) \) since any \( z = q + k \phi(p) \) can be written as \( z = q \mod \phi(p) \)
Cryptographic Strength

1. Strong Primes (i.e., Sophie-Germain) primes
   - $P = 2Q + 1$, where $P, Q =$ primes; $Q =$ Largest Prime Factor (lpf) of $P$

2. Schnorr subgroups
   - $P = kQ + 1$, where $k$ may be small
   - Generation and Validation of Group Choices
     - Estimate on 25 MHZ RISC or 66 MHZ CISC
     - Generation of $P, k, Q$ $\Rightarrow$ about 10 minutes for a group of $2^{1024}$ elements
     - Validation $\Rightarrow$ 1 minute

3. Key Length Estimates
   - practical level of security: 75 bits $\Rightarrow$ $Q =$ lpf$(P) =$ 150 bits $\Rightarrow$ $P =$ > 980 bits
   - size of exponent should be at least $2 \times$ length of key $= 2 \times 75 = 180$ bits

   - 20 year security: 90 bits $\Rightarrow$ $Q =$ lpf$(P) =$ 180 bits $\Rightarrow$ $P =$ > 1400 bits
   - size of exponent should be at least $2 \times$ length of key $= 2 \times 90 = 180$ bits

   - extended security: 128 bits $\Rightarrow$ $Q =$ lpf$(P) =$ 256 bits $\Rightarrow$ $P =$ > 3000 bits
   - size of exponent should be at least $2 \times$ length of key $= 2 \times 128 = 156$ bits

4. Reuse of $x$ (e.g., more than 100 times) $\Rightarrow$ timing attacks on $x$; use “blinding factor” $r$
   - $A = (r g^y)$, where $r$ is a random group element
   - $B = A^x = (r g^y)^x = (r^x)(g^{xy})$
   - $C = B (r^{-x}) = (r^x)(r^{-x})(g^{xy}) = g^{xy}$
Group Descriptors - 2 Examples

Group Type: MODP /* modular exponentiation group, mod P*/
Size of Field (in bits): $\left\lceil \log_2 P \right\rceil$ a 32-bit integer
Defining Prime $P$: a multi-precision integer
Generator $G$: a multi-precision integer $2 \leq G \leq P-2$
optional:
Largest prime factor of $P-1$: the multiprecision integer $Q$
Strength of Group: a 32-bit integer (approx. the no. of key bits protected; $\log_2$ of workfactor)

Group Type: ECP /* elliptic curve group, mod P */
Size of Field (in bits): $\left\lceil \log_2 P \right\rceil$ a 32-bit integer
Defining Prime $P$: a multi-precision integer
Generator $(X, Y)$: two multi-precision integers $(X, Y \leq P)$
Parameters of the curve $A, B$: two multi-precision integers $(A, B \leq P)$
optional:
Largest prime factor of group order: the multi-precision integer
Order of the group: a multi-precision integer
Strength of Group: a 32-bit integer (approx. the no. of key bits protected; $\log_2$ of workfactor)

elliptic curve equation: $Y^2 = X^3 + AX + B$