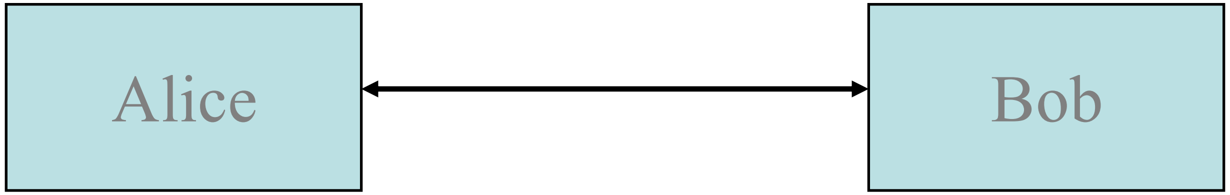
Public-Key Cryptography

The new era (1976-present)

Classical, Symmetric Ciphers

- Alice and Bob share the same secret key $K_{A,B}$.
- $K_{A,B}$ must be secretly generated and exchanged prior to using the unsecure channel.



Diffie and Hellman [1976] "New Directions in Cryptography"

Split the Bob's secret key K into two parts:

- K_E to be used for encrypting messages to Bob.
- K_D to be used for decrypting messages by Bob.
- K_E can (must) be made public
  - public key cryptography, asymmetric cryptography

"New Directions in Cryptography"

- The Diffie-Hellman paper [IEEE Transactions on Information Theory, vol. IT-22, Nov. 1976] generated lot of interest in crypto research in academia and private industry
  - http://www-ee.stanford.edu/%7Ehellman/publications/24.pdf
- Diffie & Hellman came up with the revolutionary idea of public key cryptography, but did not have a proposed implementation (this came up two years later with Merkle-Hellman and Rivest-Shamir-Adelman)
- In their '76 paper, Diffie & Hellman did invent a method for key exchange over insecure communication lines, a method that is *still in use* today

Excerpts from RSA paper

- Historical paper: "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems R.L. Rivest, A. Shamir, and L. Adleman", *Communications of the ACM* **21**, 1978
- "The era of electronic mail may soon be upon us; we must ensure that two important properties of the current paper mail system are preserved: (a) messages are private, and (b) messages can be *signed*. We demonstrate in this paper how to build these capabilities into an electronic mail system.

  At the heart of our proposal is a new encryption method. This method provides an implementation of a public-key cryptosystem, an elegant concept invented by Diffie and Hellman [1]. Their article motivated our research, since they presented the concept but not any practical implementation of such a system. Readers familiar with [1] may wish to skip directly to Section V for a description of our method."

Integer Multiplication & Factoring as a One Way Function

$$\begin{tikzcd} p,q \arrow[r, bend left, "\text{easy}"] & N=pq \arrow[l, bend left, "\text{hard}"] \end{tikzcd}$$

Q.: Can a public key system be based on this observation ?????

Discrete Log (DL)

- Let G be a finite cyclic group with n elements and g a generator of G
- Let y be any element in G; then it can be written as $y = g^x$, for some integer x
- Let $y = g^x$ and x the minimal non negative integer satisfying the equation.
- The minimal x s.t. $y = g^x$ is called the *discrete log* of y to base g.
- Example: $y = g^x$ mod p in the multiplicative group of Z_p

Quick exponentiation (C code)

```
static long fastExp(int base, int exp) {
    long f = 1;
    long b = base;
    while(exp > 0) {
        int lsb = 0x1 & exp;
        exp >>= 1;
        if(lsb) f *= b;
        b *= b;
    }
    return f;
}
```

Add mod *p* as needed

Discrete Log in Z_p: A candidate as One Way Function

- Let $y = g^x \bmod p$ in Z_p^*, the multiplicative group of Z_p (set of positive integers less than p and relatively prime to p, with multiplication - e.g. $Z_{10}^* = \{1, 3, 7, 9\}$)
- $g^x \bmod p$ can be computed in $O(\log x)$ multiplications, each multiplication taking $O(\log^2 p)$ steps (remind $g < p$)
  - also, it turns out x must be $< p \Rightarrow$ overall # of steps is $O(\log^3 p)$
- Standard discrete log is believed to be computationally hard.
- $x \rightarrow g^x \bmod p$ is easy (efficiently computable).
- $g^x \bmod p \rightarrow x$ believed hard (computationally infeasible).

- $x \rightarrow g^x \bmod p$ is believed to be a one way function.
- This is a computation based notion.

One-way functions

A function $f: \{0, 1\}^* \rightarrow \{0, 1\}^*$ is one-way if f can be computed by a polynomial time algorithm, but for every randomized polynomial time algorithm A, for every polynomial p(n) and all sufficiently large n

$$\Pr[f(A(f(x))) = f(x)] < 1/p(n)$$

assuming that x is chosen from the uniform distribution on $\{0, 1\}^n$, and the randomness of A.

By this definition, the function must be "hard to invert" in the average-case, rather than worst-case sense.

Do one-way functions exist?

- open problem
  - they are conjectured to exist
- Theorem. If a one-way function exist then P ≠ NP
  - it is not known whether P ≠ NP implies the existence of one-way functions

Public Exchange of Keys

- Goal: Two parties (Alice and Bob) who do not share any secret information, perform a protocol and derive the same shared key.
- Eve, who is listening, cannot obtain the new shared key if she has limited computational resources (i.e. in polynomial time)
  - unless P = NP

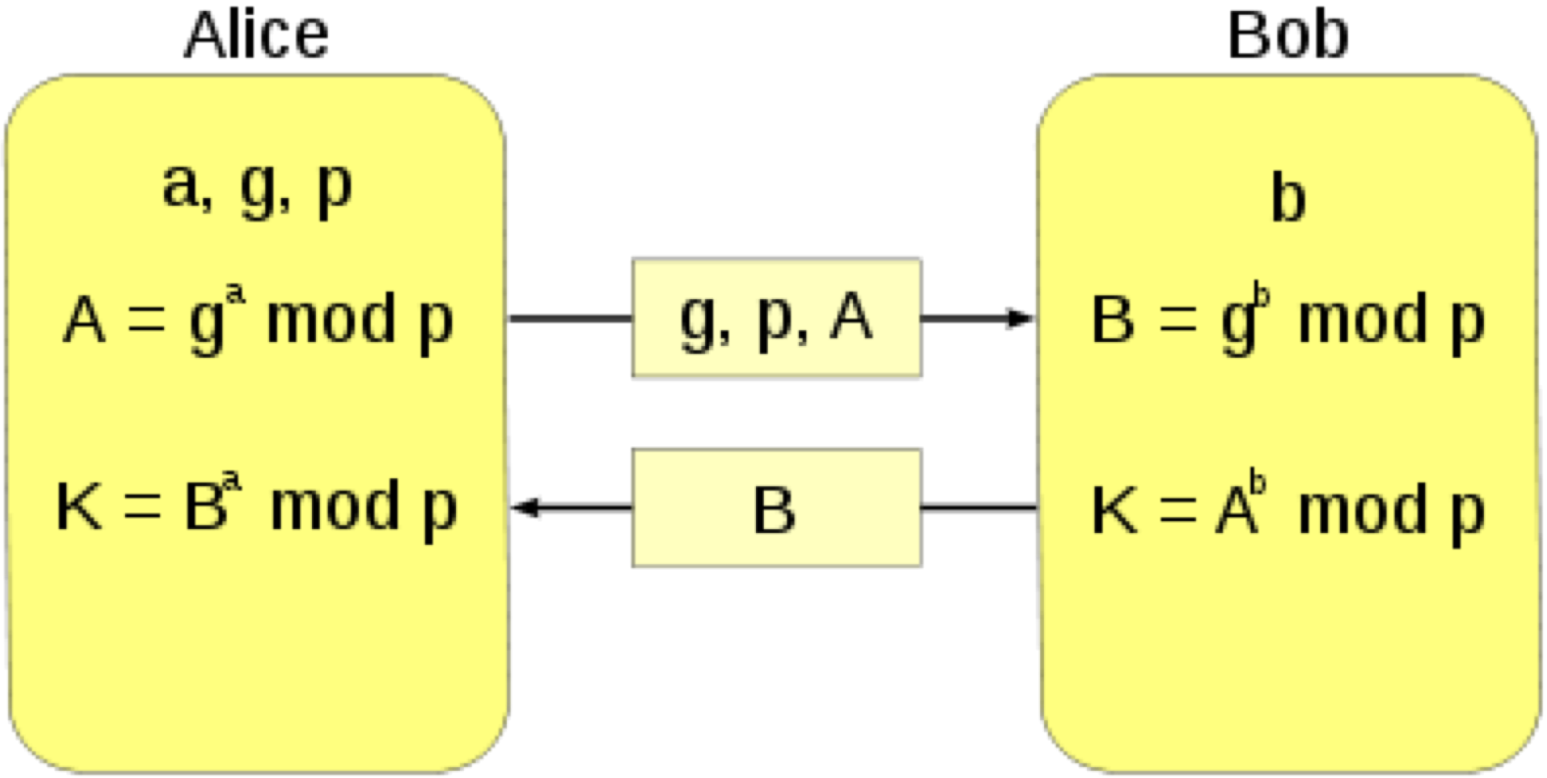
Diffie-Hellman Key Exchange

- Public parameters: a prime p, and an element **g** (possibly a generator of the multiplicative group Z_p*)
  - for technical reasons it is worthy choosing p as a safe prime, i.e. a prime p = 2q+1 where q is also prime (a Sophie Germain prime)
- Alice chooses a at random from the interval [1..p-1] and sends g^a mod p to Bob.
- Bob chooses b at random from the interval [1..p-1] and sends g^b mod p to Alice.
- Alice and Bob compute the shared key g^{ab} mod p:

Bob holds b, computes $(g^a)^b$ mod p = g^{ab} mod p .

Alice holds a, computes $(g^b)^a$ mod p = g^{ab} mod p .

Diffie-Hellman Key Exchange



$$K = A^b \bmod p = (g^a \bmod p)^b \bmod p = g^{ab} \bmod p = (g^b \bmod p)^a \bmod p = B^a \bmod p$$

DH Security

- DH is at most as strong as DL in $Z_p{}^*$.
- Formal equivalence unknown, though some partial results known.
- Despite 25 years effort, still considered secure to date.
- Secret integers a and b are discarded at the end of the session, hence Diffie–Hellman key exchange achieves ***perfect forward secrecy***, because no long-term private keying material exists to be disclosed.
- Computation time is $O(\log^3 p)$.

Weak Diffie-Hellman and the Logjam Attack

Logjam attack against the TLS protocol. The Logjam attack allows a man-in-the-middle attacker to downgrade vulnerable TLS connections to 512-bit export-grade cryptography. This allows the attacker to read and modify any data passed over the connection.

Threats from state-level adversaries. Millions of HTTPS, SSH, and VPN servers all use the same prime numbers for Diffie-Hellman key exchange. Practitioners believed this was safe as long as new key exchange messages were generated for every connection. However, the first step in the number field sieve—the most efficient algorithm for breaking a Diffie-Hellman connection—is dependent only on this prime. After this first step, an attacker can quickly break individual connections.

https://weakdh.org/
https://weakdh.org/imperfect-forward-secrecy-ccs15.pdf

Perfect forward secrecy

The cryptosystem

- generates random public keys per session for the purposes of key agreement, and
- does not use any sort of deterministic algorithm in doing so

The compromise of a shared key does not compromise former, or subsequent, shared keys

Properties of Key Exchange

- Necessary security requirement: the shared secret key is a one way function of the public and transmitted information.
- Necessary "constructive" requirement: an appropriate combination of public and private pieces of information forms the shared secret key efficiently.
- DH Key exchange by itself is effective only against a passive adversary. Man-in-the-middle attack is lethal.

Man-in-the-middle attack

- Man-in-the-middle attack (MITM), or bucket-brigade attack, or sometimes Janus attack
- The attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker.
- The attacker must be able to intercept all messages going between the two victims and inject new ones, which is straightforward in many circumstances.
- A man-in-the-middle attack can succeed only when the attacker can impersonate each endpoint to the satisfaction of the other. Most cryptographic protocols include some form of endpoint authentication specifically to prevent MITM attacks.

Security Requirements

- Is the one-way relationship between public information and shared private key sufficient?
- A one-way function may leak some bits of its arguments: this is prevented by carefully choosing g and p
  - g = generator, p = safe prime

Security Requirements (cont.)

- The full requirement is: given all the communication recorded throughout the protocol, computing *any* bit of the shared key is hard
- Note that the "any bit" requirement is especially important

Other DH Systems

- The DH idea can be used with any group structure
- Limitation: groups in which the discrete log can be easily computed are not useful (for example: additive group of Z_p)
- Currently useful DH systems: the multiplicative group of Z_p and elliptic curve systems

Key Exchange in Systems

- VPN usually has two phases
  - Handshake protocol: key exchange between parties sets symmetric keys
  - Traffic protocol: communication is encrypted and authenticated by symmetric keys
- Automatic distribution of keys- flexibility and scalability
- Periodic refreshing of keys- reduced material for attacks, recovery from leaks