Course: Modern Cryptography

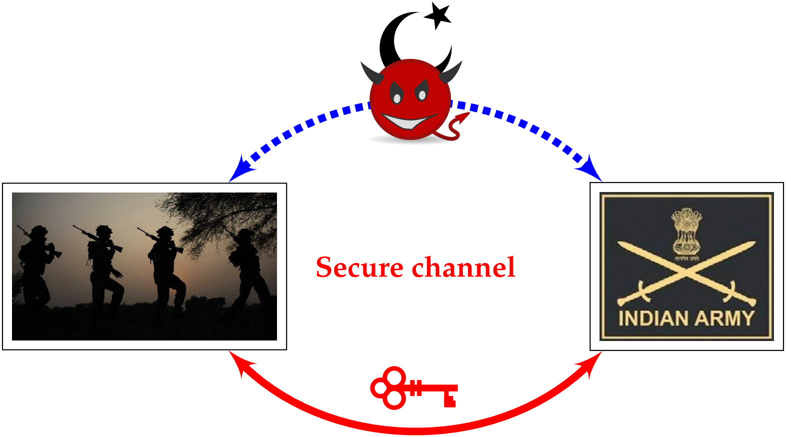
Prerequisite: Public-Key Revolution

Shashank Singh

IISER Bhopal

October 24, 2025

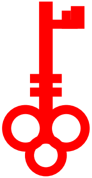
Secure Communication



110101010111111111111001000001110100011100101100011111

PRIVATE KEY CRYPTOGRAPHY

Bob



Alice

- A secret key is needed, which is shared in advance between the communicating parties, using some secured channel.
- The two parties later leverage this key to communicate securely over a public channel, using Private Key Encryption schemes.
- How does the two parties establish the secret key k?

1. Point-to-point Key Distribution

– Using a trusted courier.

– Meeting in person.

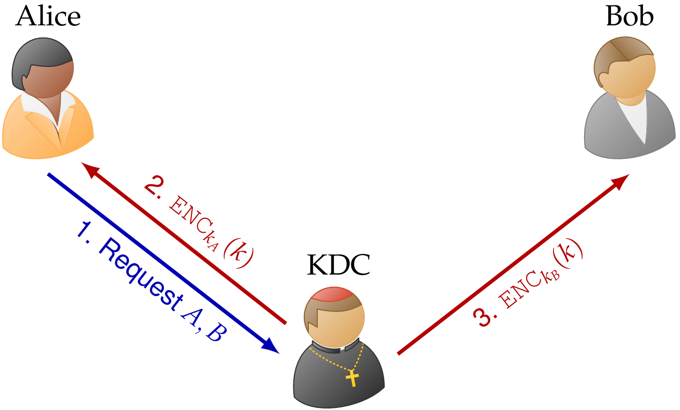
– A SIM card that contains an authentication key[1].

None of the above methods are practical for large scale application.

[1] https://cryptography101.ca/crypto101-building-blocks/

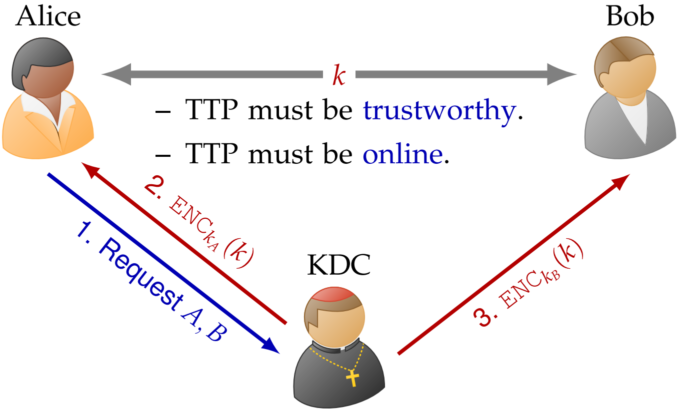
2. Using a Trusted Third Party (TTP)

A trusted third party T serve as a Key Distribution Center (KDC) and each user, say Alice, shares a secret key k_A with T.



2. Using a Trusted Third Party (TTP)

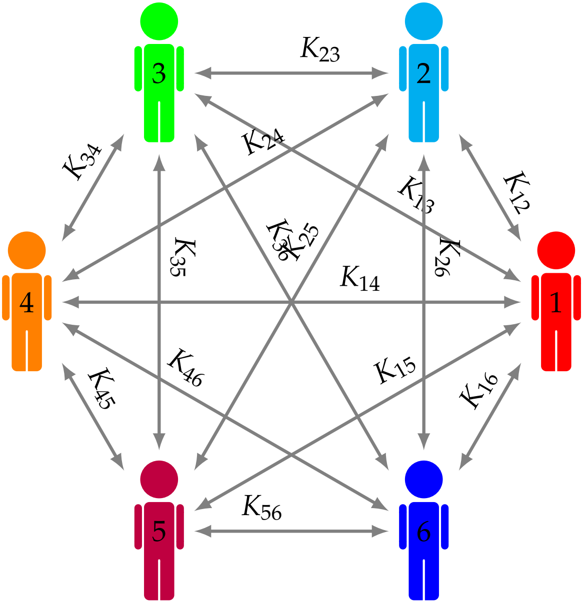
A trusted third party T serve as a Key Distribution Center (KDC) and each user, say Alice, shares a secret key k_A with T.



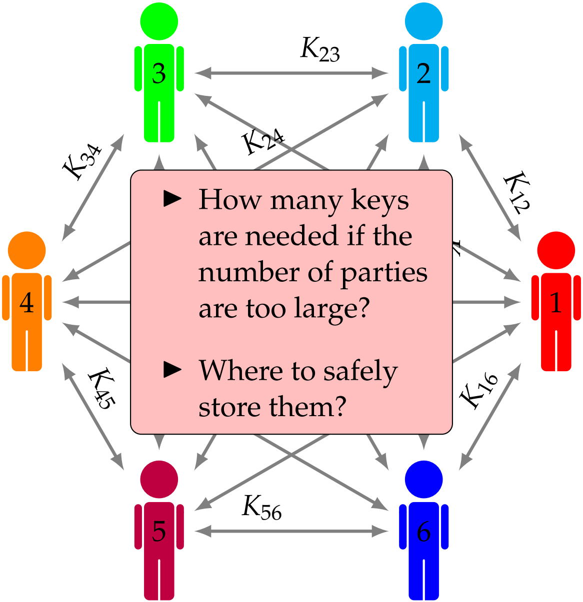
- TTP must be trustworthy.
- TTP must be online.

The TTP is a critical reliablity point and hence it is an attractive target.

Key Management Issues



Key Management Issues



- How many keys are needed if the number of parties are too large?
- Where to safely store them?

Non-repudiation is not Guaranteed

- The non-repudiation is an importation security notion.
- Non-repudiation is the property of agreeing to adhere to an obligation. It is the inability to refute previous actions or commitments.
- In case of secret key encryption schemes, it the property of sender inability to deny being the source of a message.
- Clearly, it is not guaranteed in the secret key cryptography.

- We will soon see how a simple mathematical tool helps us solve this problem.
- At this point, please take a pause and consider if it is possible to share a secret key using a channel that is completely insecure.

FINITE CYCLIC GROUPS AND DLP

- ☞ $(\mathbb{Z}/p\mathbb{Z}^*, \odot)$ is a cyclic group. Let $(\mathbb{Z}/p\mathbb{Z}^*, \odot) = \langle g \rangle$.
  - ✓ $(g, a) \to g^a$ is easy. (polynomial-time)
  - ✓ $(g, h) \to \log_g(h)$ is (computationally) hard. ((sub)exp.-time)

A Finite cyclic group: a toy Example

☞ $(\mathbb{Z}/p\mathbb{Z}^*, \odot)$ is a cyclic group.

$$\begin{aligned} p =& 1246203667817187840658350446081065904348203746516 7\\ & 8805754818788883289666801188210855036039570272508 7\\ & 4750986476843845862105486553797025393057189121768 4\\ & 3182863628469484053016144164304680668756994152469 9\\ & 31857041830305125495943713721590292853 03 \quad \text{(795-bits)}\\ g =& 5 \end{aligned}$$

☞ $(\mathbb{Z}/p\mathbb{Z}^*, \odot) = \langle g \rangle$

Exponentiation in a finite cyclic group

☞ $(\mathbb{Z}/p\mathbb{Z}^*, \odot) = \langle g \rangle$

$$\begin{aligned} h = &774356626343973985966622216\\ &006087686926705588649958206\\ &166317147722421706101723470\\ &351970238538755049093424997 \end{aligned}$$


ars TECHNICA SUBSCRIBE SIGN IN

GOT 2048 BITS? —

New crypto-cracking record reached, with less help than usual from Moore's Law

795-bit factoring and discrete logarithms achieved using more efficient algorithms.

DAN GOODIN - 12/3/2019, 7:49 PM

✓ It took Emmanuel Thomé at INRIA, France and his colleagues about 3100 CPU-years to compute $\log_g(h)$.

A TOY EXAMPLE..

☞ The value of logarithm, they got is the following.

$$\begin{aligned}\ell =&926031359281441953630949553317328555029610991914376116\\&167294204758987445623653667881005480990720934875482587\\&528029233264473672441500961216292648092075981950622133\\&6688985918666811269289825060051277283214267512441114123\\&71767375547225045851716\end{aligned}$$

> ☞ How much time does it take to compute g^{ℓ} in $\mathbb{Z}/p\mathbb{Z}$?

– about $O(795)$ multiplications modulo p.

Modular exponentiation

Square and Multiply Method

$$a^{170} \pmod{p} = a^{0b10101010} \pmod{p}$$

Modular exponentiation

Square and Multiply Method

$$a^{170} \pmod p = a^{0b10101010} \pmod p$$

$$\begin{aligned}
a^{170} &= \left(a^{85}\right)^2 &&= \left(a^{0b1010101}\right)^2 && \bmod p \\
a^{85} &= \left(a^{42}\right)^2 \cdot a &&= \left(a^{0b101010}\right)^2 \cdot a && \bmod p \\
a^{42} &= \left(a^{21}\right)^2 &&= \left(a^{0b10101}\right)^2 && \bmod p \\
a^{21} &= \left(a^{10}\right)^2 \cdot a &&= \left(a^{0b1010}\right)^2 \cdot a && \bmod p \\
a^{10} &= \left(a^{5}\right)^2 &&= \left(a^{0b101}\right)^2 && \bmod p \\
a^{5} &= \left(a^{2}\right)^2 \cdot a &&= \left(a^{0b10}\right)^2 \cdot a && \bmod p \\
a^{2} &= \left(a\right)^2 &&= \left(a^{0b1}\right)^2 && \bmod p \\
a^{1} &= 1 \cdot a &&= 1 \cdot a && \bmod p
\end{aligned}$$

Complexity?

Modular exponentiation..

Algorithm 1: Square-and-Multiply(a, n, p)

Input: a, n, p // $a = a \pmod p$; $n = n \mod (p-1)$
Output: $a^n \pmod p$
res $\leftarrow 1$
$n = (n_t \ldots n_1 n_0)_2$
for $i = t$ *to* 0 **do**
  res $\leftarrow$ res$^2 \pmod p$ // Square
  if $n_i = 1$ **then**
    res $\leftarrow$ res $\cdot\, a \pmod p$ // Multiply
return res

DLP

- There are cyclic groups (other than $(\mathbb{Z}_p^{\star}, \dot{})$), such as the group of elliptic curves over the finite fields where the time complexity of solving DLP is exponetial time.
- Hence we can assume there exists finite cyclic groups where solving DLP is infeasible.

Discrete Logarithm (Inverse of Exponentiation)

Let $E(\mathbb{F}_p) = \langle P \rangle$, where p is a prime, be a group of points of a elliptic curve over the finite field $\mathbb{F}_p$. Given $Q \in E(\mathbb{F}_p)$, it is very difficult to compute $\log_P(Q)$. For an elliptic curve group of order q, the best known algorithm to compute the value of e, requires $\mathcal{O}\left(\sqrt{q}\right)$ group operations.

$$\mathcal{O}\left(\sqrt{q}\right) \approx \left(2^{128}\right) \tag{1}$$

Diffie Hellman Problem

Diffie Hellman Problem

Given a Cryptographic group G, with a generator g and group order q,

- it is hard to compute g^{ab}, given g^a and g^b.

  - One way to do is to solve DLP first.
  - It is believed to be as hard as the DLP.

RSA Problem and Factorisation Problem

Let

$$n = p \times q \text{ and } p \neq q$$

where p and q are odd primes roughly equal to $\sqrt{n}$.

Factorisation Problem

Given n, it is computationally hard to find p or q.

RSA Problem (RSAP)

Given the group of units modulo n i.e., $(\mathbb{Z}/n\mathbb{Z})^*$ and a positive integer e such that $\gcd(e, (p-1)(q-1)) = 1$, and a random element $c \in (\mathbb{Z}/n\mathbb{Z})^*$, find an integer m such that $m^e \equiv c \pmod{n}$.

Note that

$$|(\mathbb{Z}/n\mathbb{Z})^*| = \phi(n) = (p-1)\cdot(q-1) \tag{2}$$

The RSA problem, the problem of factoring n and the problem of computing Euler's Totient $\phi(n)$ are computationally equivalent.

The state-of-the-art algorithm for factoring n is the Number Field Sieve (NFS) algorithm.