element of a finite field $F_p$.

For $g^n = a$, given $a, c$ find $n$.

$g^x = c$, given $g, c, \text{find } x$.

Primality test problem over a finite field. bisected log problem over a finite field.

not prime $\rightarrow$ composites $\rightarrow$ n = $\phi(p)$.

must be prime

$n, \phi(n)$ problem (modular generator).

Diffie-Hellman.
Solving the discrete log ---

Chinese Remainder Theorem

Can be decomposed to smaller problems

1 <= p_i * q_i <= n

in the # of bits in n

mod p_i
- Southern Stream
- western side

A set of keys

\[ A \xrightarrow{B} B \xrightarrow{A} \]

- A triple
- 1st primitive key algebra

- Diffie-Hellman
- with attacker not easy
- Construction is hard

- From middle
Plaing

\[
\text{RSA} - \quad \text{Plaing}
\]

\[
\text{versa}
\]

\[
\text{From } k_2 \text{ & vice}
\]

\[
\text{not denoted}
\]

\[
k_1 \neq k_2
\]
Choose a large prime number $n = p \cdot q$ (modulus)

Compute $e(n) = (p-1)(q-1)$

Compute the multiplicative inverse of $e$ modulo $(p-1)(q-1)$

Let $d$ be the multiplicative inverse of $e$ such that $e \cdot d \equiv 1 \pmod{(p-1)(q-1)}$

$\text{RSA} - \text{Public, Shveir, Adleman}$
\[(p, n) \text{ private key}

\left[ \begin{array}{c}
\text{d} \\
\text{e} = \text{mod } (n)
\end{array} \right] = 1

\text{Compute } d \text{ such that } d \cdot e \equiv 1 \text{ mod } \varphi(n)

\text{public key } = (e, n)
RSA simplifies (plaintext) mod n to cipher

For each p ∈ \{1, 123 \} 678

\[ x \rightarrow b \mod \ell \]

\[ y = \ell \times b \mod \ell \]

\[ \ell = p \times q \]

Saw

decrypted ← (cipher)

\[ \text{med n} \rightarrow \text{cipher} \]
Which can you do with your car?
2. Key pairs — modulus will be diff

\[ \text{Priv key} = \{ (p, \phi(p), d) \} \]

\[ \text{Pub key} = \{ e \} \]

\[ \text{msg}^e \mod p \]
Sent Communications
Authentication

\[ \text{[no password]} \]

\[ L_2 \rightarrow L_1 \]

Response

\( f = E_{k_2}(r) \)

Challenge

A

B

Verify

and set

\( f \)
Consider a graph $G$ with $n$ vertices. Let $L_1$ be the set of all edges in $G$. For each edge $e_i$ in $L_1$, if $e_i$ is not in $L_2$, then $e_i$ is added to $L_2$. After adding all such edges, we have $L_2$. This process is repeated for a specific number of times. Eventually, we reach a state where no new edges can be added to $L_2$. After this step, we set $A = L_1 \cup L_2$.
Bob gets a mum from Alice

Bob encrypts this mum & sends the result to Alice
dangerous, do NOT do
\[ \mathcal{M} \subseteq \mathbb{R}^{2m} \]

\[ \mathcal{E}(\mathcal{M}) \]

\[ \mathcal{R} \]

\[ \mathcal{K} \]

\[ \mathcal{L} \]

\[ \mathcal{O} \]
μ
→

\text{Sign}

\text{Product Key}

\text{Everywhere}

\text{Depressed}

\text{Depressed Signatures}
Digital Certificate

DOC $\rightarrow$ hash $\rightarrow$ Signature of 
Cert Authority $\rightarrow$ CA