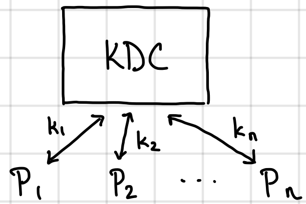
Thus far, we have <u>assumed</u> that parties have a <u>shared</u> key. Where does the shared key come from?

<u>Approach 1</u>: have a key-distribution center (KDC)



shared key between KDC and each party P_i

if P_i wants to talk to P_j:

- P_i sends nonce r_i (replay prevention) and identifier id_i to P_j
- P_j chooses nonce r_j and identifier id_j to P_i and KDC
- KDC samples k_{ij} and gives

often called a "ticket"

$$\left.\begin{aligned} c_i &\leftarrow \text{Enc}(k_{i,\text{Enc}}, k_{ij}) \\ t_i &\leftarrow \text{MAC}(k_{i,\text{MAC}}, (r_i, r_j, id_i, id_j, c_i)) \end{aligned}\right\} \text{to } P_i$$

$$\left.\begin{aligned} c_j &\leftarrow \text{Enc}(k_{j,\text{Enc}}, k_{ij}) \\ t_j &\leftarrow \text{MAC}(k_{j,\text{MAC}}, (r_i, r_j, id_i, id_j, c_j)) \end{aligned}\right\} \text{to } P_j$$

nonces needed to ensure "freshness" for session (no replays) and identifiers needed to bind session key k_{ij} to identities id_i, id_j

Basic design for Kerberos — only requires symmetric primitives

- <u>Drawback</u>: KDC must be fully trusted (knows everyone's keys) and is <u>single</u> point of failure (no session setup if KDC goes offline!)

Public-key cryptography: Session setup / key-exchange without a KDC

Diffie-Hellman key exchange (example) — will be more precise later:

- Assume we have a fixed prime p and a value $g \in \{1, 2, \ldots, p-1\}$ (these could be specified in a cryptographic standard)

Alice		**Bob**
$x \xleftarrow{R} \{1, 2, \ldots, p-1\}$		$y \xleftarrow{R} \{1, 2, \ldots, p-1\}$
	$\xrightarrow{g^x \pmod p}$	
	$\xleftarrow{g^y \pmod p}$	
compute $g^{xy} = (g^y)^x \pmod p$		compute $g^{xy} = (g^x)^y \pmod p$

shared secret: g^{xy}

- Assumption: given only (g, p), g^x, g^y, it is difficult to compute g^{xy} [computational Diffie-Hellman assumption]
  - ↳ better be the case that computing logarithms base-g be difficult [discrete logarithm problem] (e.g., given g, g^x, cannot compute x)

To understand this more broadly, we will need some math background. We discuss some key facts from number theory and abstract algebra below:

Definition. A group consists of a set $\mathbb{G}$ together with an operation $*$ that satisfies the following properties:

- Closure: If $g_1, g_2 \in \mathbb{G}$, then $g_1 * g_2 \in \mathbb{G}$
- Associativity: For all $g_1, g_2, g_3 \in \mathbb{G}$, $g_1 * (g_2 * g_3) = (g_1 * g_2) * g_3$
- Identity: There exists an element $e \in \mathbb{G}$ such that $e * g = g = g * e$ for all $g \in \mathbb{G}$
- Inverse: For every element $g \in \mathbb{G}$, there exists an element $g^{-1} \in \mathbb{G}$ such that $g * g^{-1} = e = g^{-1} * g$

In addition, we say a group is commutative (or abelian) if the following property also holds:

- Commutative: For all $g_1, g_2 \in \mathbb{G}$, $g_1 * g_2 = g_2 * g_1$

called "multiplicative" notation

Notation: Typically, we will use "$\cdot$" to denote the group operation (unless explictly specified otherwise). We will write g^x to denote $\underbrace{g \cdot g \cdot g \cdots g}_{x \text{ times}}$ (the usual exponential notation). We use "1" to denote the multiplicative identity.

Examples of groups: $(\mathbb{R}, +)$: real numbers under addition

$(\mathbb{Z}, +)$: integers under addition

$(\mathbb{Z}_p, +)$: integers modulo p under addition [sometimes written as $\mathbb{Z}/p\mathbb{Z}$]

here, p is prime

The structure of $\mathbb{Z}_p^*$ (an important group for cryptography):

$\mathbb{Z}_p^* = \{x \in \mathbb{Z}_p : \text{there exists } y \in \mathbb{Z}_p \text{ where } xy = 1 \pmod p\}$

↖ the set of elements with multiplicative inverses modulo p