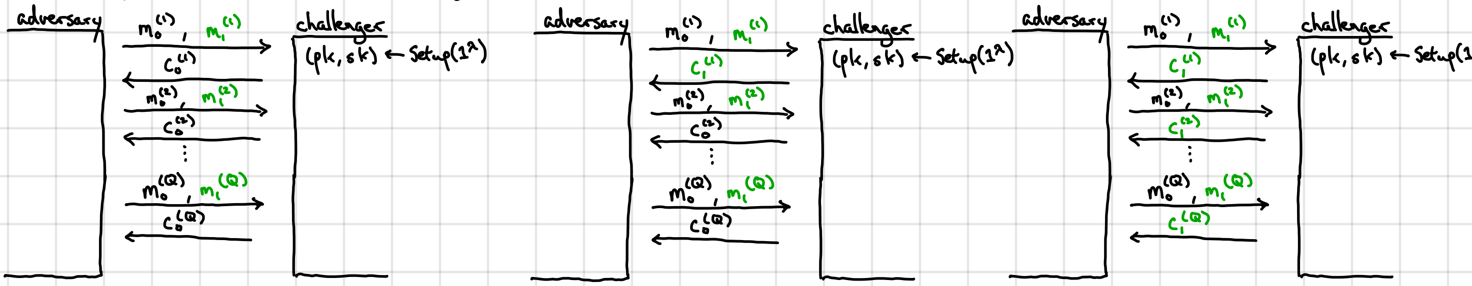
In the secret-key setting, we distinguished between semantic security and CPA-security. Here, this is <u>unnecessary</u> since

semantic security $\Rightarrow$ CPA security [means that public-key encryption must be randomized!]

↳ <u>Intuitively</u>: adversary can encrypt messages on its own (using the public key)

<u>Formally</u>: Follows from a hybrid argument



$b = 0$ [always encrypt m_0] Intermediate experiments $b = 1$ [always encrypt m_1]

Total of $Q-1$ intermediate distributions

↳ i^{th} distribution and $(i+1)^{\text{st}}$ distribution identical except on $(m_0^{(i)}, m_1^{(i)})$, challenger encrypts $m_0^{(i)}$ in distribution i and $m_1^{(i)}$ in distribution $i+1$

↳ these two distributions are indistinguishable by <u>semantic security</u> [in the reduction, the encryptions of the other messages (index $\neq i$) can be constructed using the public key (and do not depend on the challenger's choice bit)]

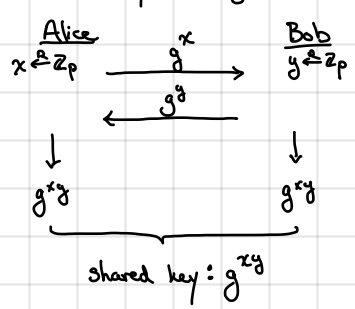
↳ if an adversary can distinguish endpoints $(b=0, b=1)$, then it must be able to distinguish a pair of intermediate distributions [by triangle inequality]

$\therefore$ semantic security $\Rightarrow$ every pair of distributions is computationally indistinguishable

$\Rightarrow$ CPA-security

<u>PKE from DDH (ElGamal)</u>: Let $\mathbb{G}$ be a group with generator g and prime order p

Recall Diffie-Hellman key exchange:



<u>Idea</u>: Alice will publish $h = g^x$ as her public key

Bob encrypts by choosing fresh share g^y and uses g^{xy} to encrypt the message

security parameter dictates what group is used (eg, P-256, P-384, P-512)

$\text{Setup}(1^\lambda)$: $x \xleftarrow{R} \mathbb{Z}_p$ $\quad$ pk: h $\quad$ $\mathcal{M} = \mathbb{G}$

$h \leftarrow g^x$ $\quad$ sk: x $\quad$ $\mathcal{C} = \mathbb{G}^2$

$\text{Encrypt}(\overset{=h}{pk}, m)$: $y \xleftarrow{R} \mathbb{Z}_p$

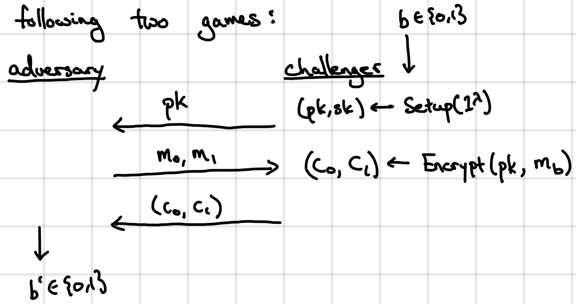
$c \leftarrow (g^y, m \cdot h^y)$

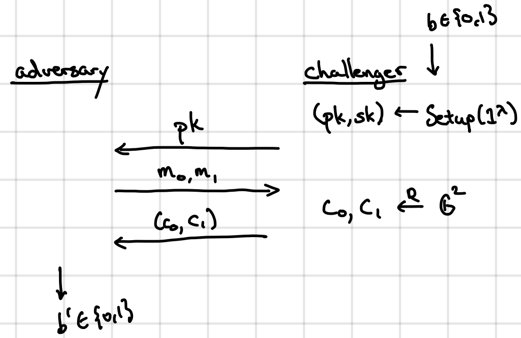
$\text{Decrypt}(\overset{=x}{sk}, c)$: $m \leftarrow c_1 / c_0^x$

<u>Correctness</u>: $\dfrac{c_1}{c_0^x} = \dfrac{m \cdot h^y}{(g^y)^x} = \dfrac{m \cdot (g^x)^y}{(g^y)^x} = \dfrac{m \cdot g^{xy}}{g^{xy}} = m$

Security: If DDH holds in $\mathbb{G}$, then ElGamal is semantically secure.

Proof. Consider following two games:

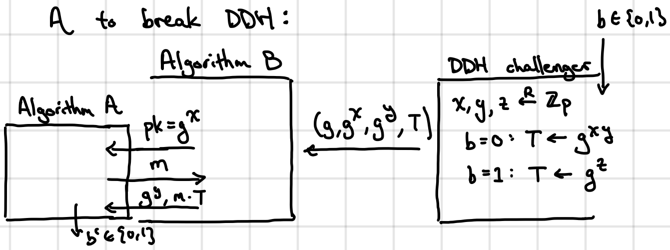




Claim: these two games are indistinguishable under DDH

adversary's advantage in guessing b is 0 here since (c_0, c_1) is independent of (m_0, m_1)!

Proof. Suppose there exists efficient A that can distinguish $(c_0, c_1) \leftarrow \mathrm{Encrypt}(pk, m)$ from $(c_0, c_1) \xleftarrow{R} \mathbb{G}^2$. We use A to break DDH:



Observe: x is uniform over $\mathbb{Z}_p$ so g^x is a properly-generated public key (for ElGamal)

if $T = g^{xy}$, then $(g^y, T \cdot m) = (g^y, g^{xy} \cdot m)$ which is the output of $\mathrm{Encrypt}(pk, m)$ with randomness y — this is exactly the distribution where A sees $\mathrm{Encrypt}(pk, m)$

if $T = g^z$, then $(g^y, g^z \cdot m)$ is uniform over $\mathbb{G}^2$ (since y, z are sampled independently of each other and of m) — this is exactly the distribution where A sees $(c_0, c_1) \xleftarrow{R} \mathbb{G}^2$

distinguishing advantage of B = distinguishing advantage of A

Equivalent view: Under DDH, g^{xy} looks uniform even given g, g^x, g^y, so an ElGamal ciphertext looks indistinguishable (to an efficient adversary) from a OTP encryption

What if we want to encrypt longer messages? [or messages that is not a group element]

- Hybrid encryption (key encapsulation [KEM]):
  - Use PKE scheme to encrypt a secret key
  - Encrypt payload using secret key + authenticated encryption

called key encapsulation

PKE. Encrypt (pk, k)	"header"	[slow]
AE. Encrypt (k, m)	"payload"	[fast]

secret-key operations much much faster than public-key operations!

- How to derive key from group element?

  Same as in key-exchange: hash the group element to a bit-string (symmetric key)

  e.g., Hash-ElGamal: $\mathrm{Encrypt}(pk, m)$: $y \xleftarrow{R} \mathbb{Z}_p$

  $$c = (g^y, m \oplus H(g, h, g^y, h^y))$$

  as before, can also rely on CDH + ideal hash function (random oracle)

  $H: \mathbb{G}^4 \to \{0,1\}^n$

Vanilla ElGamal described above is not CCA-secure!

Ciphertexts are malleable: given $ct = (g^y, h^y \cdot m)$, can construct ciphertext $(g^y, h^y \cdot m \cdot g)$ which decrypts to message $m \cdot g$

↳ directly implies a CCA attack

Several approaches to get CCA security from DH assumptions:

- Cramer-Shoup (CCA-security from DDH) - based on hash-proof systems
- Fujisaki-Okamoto transformation (using an ideal hash function + CDH)
- Make stronger assumption ("interactive" CDH + use ideal hash function):
  - Setup(1^λ): $x \xleftarrow{R} \mathbb{Z}_p$ $\quad$ pk: h
    $h \leftarrow g^x$ $\quad$ sk: x
  - Encrypt(pk, m): $y \xleftarrow{R} \mathbb{Z}_p$ $\quad k \leftarrow H(g, g^x, g^y, h^y)$ $\quad ct' \leftarrow \text{Enc}_{AE}(k, m)$
    $c \leftarrow (g^y, ct')$
  - Decrypt(sk, c): $k \leftarrow H(g, g^x, c_0, c_0^x)$
    $m \leftarrow \text{Dec}_{AE}(k, c_1)$

Essentially ElGamal where key derived from hash function

also called strong DH assumption

symmetric authenticated encryption scheme

We do not know of any groups where CDH believed to be hard, but interactive CDH is easy.

"CDH is hard even given access to a DDH oracle"

Elliptic-curve groups: a candidate group where the best known discrete log algorithms are the generic ones
↳ Studied by mathematicians since antiquity! [See work of Diophantus, circa 200 AD]
↳ Proposed for use in cryptographic applications in the 1980s ⟶ now is a leading choice for public-key cryptography on the web [another example where abstract concepts in mathematics end up having surprising consequences]

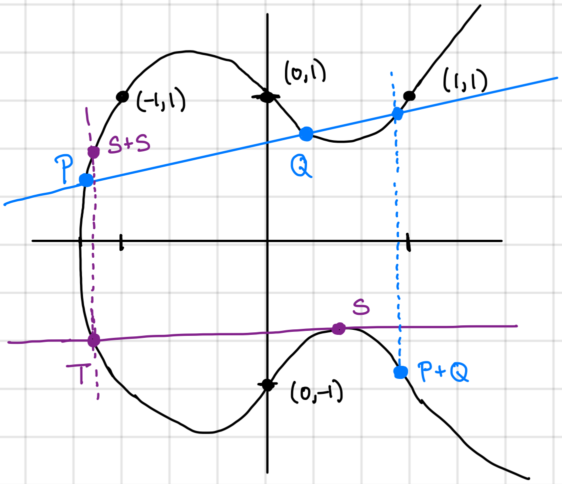
An elliptic curve is defined by an equation of the following form:

$$E: y^2 = x^3 + Ax + B$$

[we will assume that $\underbrace{4A^3 + 27B^2}_{\text{"discriminant" of the curve}} \neq 0$] non-zero to ensure there are no repeated roots (and the group law is well-defined)

where A, B are constants (over $\mathbb{R}$ or $\mathbb{C}$ or $\mathbb{Q}$ or $\mathbb{Z}_p$)

Example of an elliptic curve: $y^2 = x^3 - x + 1$ (over the reals)



Consider the set of rational points on this curve (points where x- and y-coordinates are rational values)

e.g., $(0, \pm 1)$, $(1, \pm 1)$, $(-1, \pm 1)$ [are there other points?]

Surprising facts:

1. Take any two rational points on the curve and consider the line that passes through them. The line will intersect the curve at a new point, which will also have rational coefficients.
2. Take any rational point on the curve and consider the tangent line through that point. The line will intersect the curve at a new point, which will also have rational coefficients.

Thus, given two rational points, there is a way to generate a third rational point.
↳ In fact, this operation essentially defines a group law (but with following modifications):
1. We introduce a "point at infinity" (e.g., a horizontal line at $y = \infty$), denote $\mathcal{O}$ (this is the identity element)
2. The group operation (called the "chord and tangent" method) maps two curve points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ to a point R by first computing the third point that along the line connecting P, Q and reflecting the point about the x-axis. [Observe that the reflection ensures that $\mathcal{O}$ is the identity)

↳ Remarkably, this defines a group law on the rational points on the elliptic curve, and we can write down algebraic relations for computing the group law (somewhat messy but there is a closed form expression)

In cryptography, we work over finite domains, so we instead consider elliptic curves over $\mathbb{Z}_p$ (rather than $\mathbb{R}$ or $\mathbb{C}$).
Specifically, we write

$$E(\mathbb{Z}_p) = \{x, y \in \mathbb{Z}_p : y^2 = x^3 + Ax + B\} \cup \{\mathcal{O}\}$$

No geometric interpretation of the group law over $\mathbb{Z}_p$ (instead, define it using the algebraic definitions derived above)
↳ $E(\mathbb{Z}_p)$ still forms a group under this group law

How big is the group $E(\mathbb{Z}_p)$?

Theorem (Hasse). Let E be an elliptic curve with coefficients in $\mathbb{Z}_p$. Then

$$\left| \, |E(\mathbb{Z}_p)| - (p+1) \right| \leq 2\sqrt{p}$$

Thus, number of points on $E(\mathbb{Z}_p)$ is roughly $p \pm \sqrt{p}$