Computational particles: in the following, let 6 be a finite cyclic group generated by g with order g
Theoretic by problem: sample
$$x \in \mathbb{Z}_{4}$$

given $h = g^{x}$, compute x
- Computational Diffie-Hellman (CDH): sample $x, y \notin \mathbb{Z}_{4}$
given g^{x}, g^{y} , compute g^{xy}
- Decisional Diffie-Hellman (DDH): sample $x, y, f \cong \mathbb{Z}_{4}$
distinguish between $(g, g^{x}, g^{y}, g^{y}, g^{x})$ us. $(g, g^{x}, g^{y}, g^{y}, g^{z})$
Each of these problems translates to a corresponding computational assumption:
Each of these problems translates to a corresponding computational assumption:
Deficition. Let $G = (g)$ be a finite cyclic group of order g (observe g is a function of the security parameter λ)
The DDM assumption holds in G if for all efficient adversaries $A :$
 $P_{[X, y]} \stackrel{e}{=} \mathbb{Z}_{p} : A(y, y^{x}, g^{y}, g^{x}) = 2] - P_{[X, y, f]} \stackrel{e}{=} \mathbb{Z}_{q} : A(y, g^{x}, g^{y}, g^{z}) = 1]| = negl(\lambda)$
The discurption holds in G if for all efficient adversaries A :
 $P_{[X, y]} \stackrel{e}{=} \mathbb{Z}_{g} : A(y, y^{x}, g^{y}) = g^{x}] = negl(\lambda)$
The discurption holds in G if for all efficient adversaries A :
 $P_{[X, y]} \stackrel{e}{=} \mathbb{Z}_{g} : A(y, y^{x}, g^{y}) = g^{x}] = negl(\lambda)$
Certainly : if DDH holds in G \Rightarrow CDH holds in G \Rightarrow discrete log holds in G

Diffie-Hellman key exchange

$$\begin{array}{ccc} \underline{Alice} & \underline{Bob} \\ \chi \stackrel{\text{\tiny \ensuremath{\mathbb{Z}}}}{=} & \chi \stackrel{\text{\tiny \\ensuremath{\mathbb{Z}}}}{=} & \chi \stackrel{\text{\tiny \\ensuremath{\mathbb{Z}}}}{$$

$$Compute g^{Xy} = (g^{X})^{X} \qquad compute g^{Xy} = (g^{X})^{y}$$

> shared secret:
$$g^{\chi g} \leftarrow$$

But usually, we want a random bit-string as the key, not random group element

- L> Element gxy has log p bits of entropy, so should be able to obtain a rondom bitstring with l < log p bits L> Solution is to use a "randomness extractor"
 - is Information-theoretic constructions based on universal hashing / pairwise-independent hashing
 - (loses some bits of entropy)

	L>	Use	٥-	" ra	ndom	. 000	ucle ¹¹	٥r	an	" .de	ral	hash	fun	" ition;	[Hewin	<u>ه <i>اند</i></u> :	Shr	1-256	. (ع,	<u>ع</u> ×,	9 ⁸ , 5	a ^x y)]	bind th	e enti	key .	l a
		(ver	y ef	fice	nt in	n pra	.ctrce)								ງື	rg br	actice	: had	n <u>all</u>	ing	uts \	\mathcal{I}		ι +	onscri	pt	J
			\rightarrow	Argu	ing	secul	ity ∶	1.	Re	ly or	\ H	ashDt) ass	scum p	tion.	(ر م [×] ,	g°,	H(ع,	5 ^{7,} 3	′, ე ^{ჯყ}) ≈້	(q,	<u>م</u> ,	<u>م</u> *,	r)		
					U		/			,			when	و	H : (ե ⁴ →	f0,13	n o	.nd	r &	۰,۱ وا	3^	J	Ū	0			
								2.	Μ	odel	Η	as i	deal	hask	L fu	ructi su	(H	: 6 [#] -	⇒ {0,	13 [°]	(i.	e., na	ndom	م ره	سطع)	an	7	
									16	y o	n (DH	in	ß	[in	ability	to e	valuate	, ң	on g	63	⇒	outp	ut i	s rou	dom	strin]
										1										0			,					,
Instantiation	<u>s</u> :	Disc	rete	log	in i	ℤ ۴	when	P	6	2048	8-bit	s pr	ovides	مه	proxim	wtely	128-	bits	of s	ودسة	່າຂໍ້	ر د ا						

→ Best attack is General Number Field Sieve (GNFS) - runs in time 2 time Much better than brute force - 2^{log} P → Need to choose p carefully having small prime factors if we want to double security, (e.g., avoid cases where p-1 is smooth) for DDH applications, we usually set p = 2g+1 where g is also a prime (p is a "safe prime") and work in the Scale linearly (or work) in of security)

subgroup of order g in \mathbb{Z}_{p}^{*} (\mathbb{Z}_{p}^{*} has order p-1=2g) bit length of the modulus

E١	iptic	(UL10	z grou	ps :	only	requi	ہو ،	256	-bit	modu	eula	for	12	8 bi	ts	of	secu	хèч								
	, Ļş	Best	attac	k is	gen	بمزد	atto	nck	and	runs	in	ti ,	ne.	2 2)P/2	•	ſP	- alge	rithm	~ -	car	. discuss	at.	end	of]	
	Ь	Much	faste	r tho	n W	sina	Z*	: ;	seven-l	sta	ndard	اع					Ľ	U					ŧ	jemest	ور ا	
		-	- NIS	т Р	256	, P3	84,	PSI	2			}	can	discu	s	more	at	end	of	zemi	uster					
		-	- Dan	Berns	tein's	cus	rves:	C	urve	2551	٩	J		lor ia	~ ~	. dvano	ed (crypte	, d)						
	╘⇒	Wideh	y used	for	key-	exch	ange	+	Sign	atures	5 Dr	. fha	_ w	elo												

When describing apprographic constructions, we will work with an abstract group (easier to work with, less destuils to worry about)