- Hash Diffie-Hellman Assumption

G: finite cyclic group of order n , $H: G^2 \longrightarrow K$ a hash function

Def: Hash-DH (HDH) assumption holds for (G, H) if:

$$\boxed{\begin{array}{c}(g,\ g^a,\ g^b,\ H(g^b, g^{ab})) \approx_p (g,\ g^a,\ g^b,\ R) \\[1em] \text{where } g \leftarrow \{\text{generators of } G\},\quad a, b \leftarrow Z_n,\quad R \leftarrow K\end{array}}$$

CDH is easy G $\Longrightarrow$ HDH is easy in (G, H) $\forall H$, $|Im(H)| \geq 2$

- HDH is a stronger assumption

- Example

Suppose $K = \{0,1\}^{128}$ and

$H: G^2 \longrightarrow K$ only outputs strings in K that begin with 0

(i.e. for all x,y: msb(H(x,y))=0)

Can Hash-DH hold for (G, H) ?

- ○ Yes, for some groups G
- $\Rightarrow$ ○ No, Hash-DH is easy to break in this case
- ○ Yes, Hash-DH is always true for such H

- H acts as an extractor: strange distribution on g squared => uniform on K
- very easy to distinguish the distributions
- msb of the right will be 0 with probability 1/2
- msb of the left will be 0 always

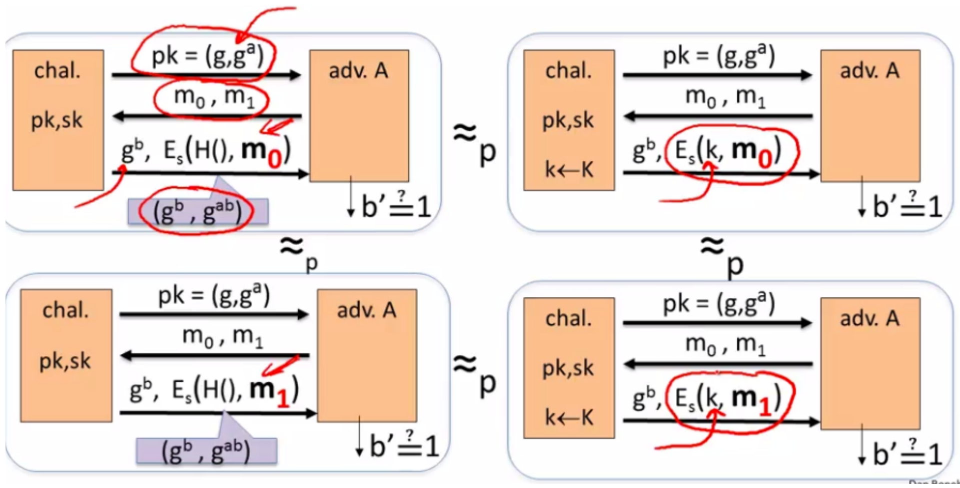
- ElGamal is semantically secure under Hash-DH

KeyGen: $g \leftarrow \{\text{generators of } G\}$, $a \leftarrow Z_n$

output $pk = (g,\ h = g^a)$, $sk = a$

E(pk=(g,h), m) :	**D(sk=a, (u,c)) :**
$b \leftarrow Z_n$	
$k \leftarrow H(g^b, \overset{g^{ab}}{\overset{\shortparallel}{h^b}})$, $c \leftarrow E_s(k, m)$	$k \leftarrow H(u, u^a)$, $m \leftarrow D_s(k, c)$
output (g^b, c)	output m

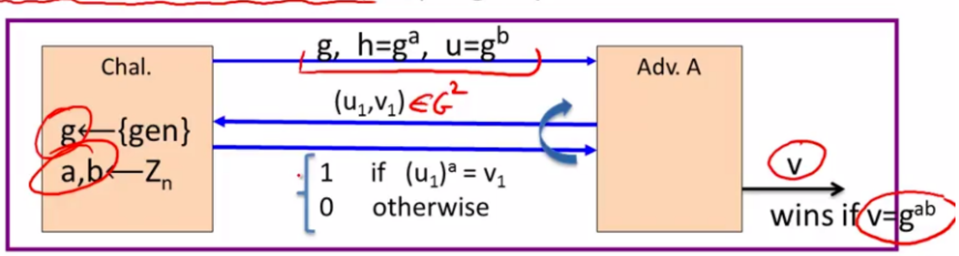
- ElGamal is semantically secure under Hash-DH
  - the output of the hash function g to the b and g to the ab is indistinguishable from random
  - if we replace the hash function by a truly random key K then the attacker cannot distinguish these two games



  - the games on the right are a symmetric encryption system and semantically secure so the two games are indistinguishable therefor the two games on the left are also computationally indistinguishable for the same reasoning.
- ElGamal chosen ciphertext security?
  - give the attacker more power => stronger assumption
  - give the attacker the ability to make queries

To prove chosen ciphertext security need stronger assumption

Interactive Diffie-Hellman (IDH) in group G:



IDH holds in G if: **∀efficient A: Pr[A outputs g^{ab}] < negligible**

- ElGamal chosen ciphertext security?

Security Theorem:

If **IDH** holds in the group G, **(E_s, D_s)** provides auth. enc.

and **H:** $G^2 \longrightarrow K$ is a "random oracle"

then **ElGamal** is CCA^{ro} secure.

- **Variants of ElGamal With a Better Security Analysis**
- Review: ElGamal encryption
  - Keygen - picks a random generator
  - a - picks a random exponent from Z n
  - output
    - pk - generator and h = generator to the a
    - sk - a
  - Encryption
  - Decryption

KeyGen: $g \leftarrow$ {generators of G} , $a \leftarrow Z_n$

output pk = $(g, h=g^a)$, sk = a

E(pk=(g,h), m) : $b \leftarrow Z_n$

$k \leftarrow H(g^b, \overset{g^{ab}}{h^b})$, $c \leftarrow E_s(k, m)$

output (g^b, c)

D(sk=a, (u,c)) :

$k \leftarrow H(u, u^a)$, $m \leftarrow D_s(k, c)$

output m

- ElGamal chosen ciphertext security

Security Theorem:

If **IDH** holds in the group G, (E_s, D_s) provides auth. enc.
and **H:** $G^2 \longrightarrow K$ is a "random oracle"
then **ElGamal** is CCA^{ro} secure.

Can we prove CCA security based on CDH $(g, g^a, g^b \not\rightarrow g^{ab})$?

- Option 1: use group G where CDH = IDH (a.k.a bilinear group)
- Option 2: change the ElGamal system

- Variants: twin ElGamal

KeyGen: $g \leftarrow$ {generators of G} , $a1, a2 \leftarrow Z_n$

output pk = $(g, h_1=g^{a1}, h_2=g^{a2})$, sk = (a1, a2)

E(pk=(g,h_1,h_2), m) : $b \leftarrow Z_n$

$k \leftarrow H(g^b, h_1{}^b, h_2{}^b)$

$c \leftarrow E_s(k, m)$

output (g^b, c)

D(sk=(a1,a2), (u,c)) :

$k \leftarrow H(u, \underset{=g^{ba_1}}{u^{a1}}, \underset{=g^{ba_2}}{u^{a2}})$

$m \leftarrow D_s(k, c)$

output m

- Chosen ciphertext security

Security Theorem:

If **CDH** holds in the group G, **(E_s, D_s)** provides auth. enc.
and **H:** $G^3 \longrightarrow K$ is a "random oracle"
then **twin ElGamal** is CCA^{ro} secure.

Cost: one more exponentiation during enc/dec

– Is it worth it? No one knows …

- ElGamal security w/o random oracles?

Can we prove CCA security without random oracles?

- Option 1: use Hash-DH assumption in "bilinear groups"
  - Special elliptic curve with more structure [CHK'04 + BB'04]

- Option 2: use Decision-DH assumption in any group [CS'98]

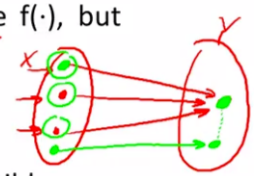
- **A unifying Theme**
- One-way functions (informal)

A function $f: X \longrightarrow Y$ is one-way if

- There is an efficient algorithm to evaluate $f(\cdot)$, but
- Inverting f is hard:

  for all efficient A and $x \leftarrow X$:

  $$\Pr[f(A(f(x))) = f(x)] < \text{negligible}$$



Functions that are not one-way: $f(x) = x$, $f(x) = 0$

- Example 1: generic one-way functions

Let $f: X \longrightarrow Y$ be a secure PRG (where $|Y| \gg |X|$)

(e.g. f built using det. counter mode)

Lemma: f a secure PRG $\Rightarrow$ f is one-way

Proof sketch:

A inverts f $\Rightarrow$ $B(y) = \begin{cases} f(A(y)) = y & \text{output } 0 \\ \text{output } 1 & \text{otherwise} \end{cases}$ is a distinguisher

$y \in Y$

Generic: no special properties. Difficult to use for key exchange.

- seed causes the generator to output the same strings

- Example 2: The DLOG one-way function

Fix a finite cyclic group G (e.g $G = (Z_p)^*$) of order n

g: a random generator in G (i.e. $G = \{1, g, g^2, g^3, \dots, g^{n-1}\}$)

Define: $f: Z_n \longrightarrow G$ as $\boxed{f(x) = g^x \in G}$

Lemma: Dlog hard in G $\Rightarrow$ f is one-way

Properties: $f(x), f(y) \Rightarrow f(x+y) = f(x) \cdot f(y)$ $\in G$

$= g^{x+y} = g^x \cdot g^y$

$\Rightarrow$ key-exchange and public-key encryption.

Dan Bon

- Example 3: The RSA one-way function

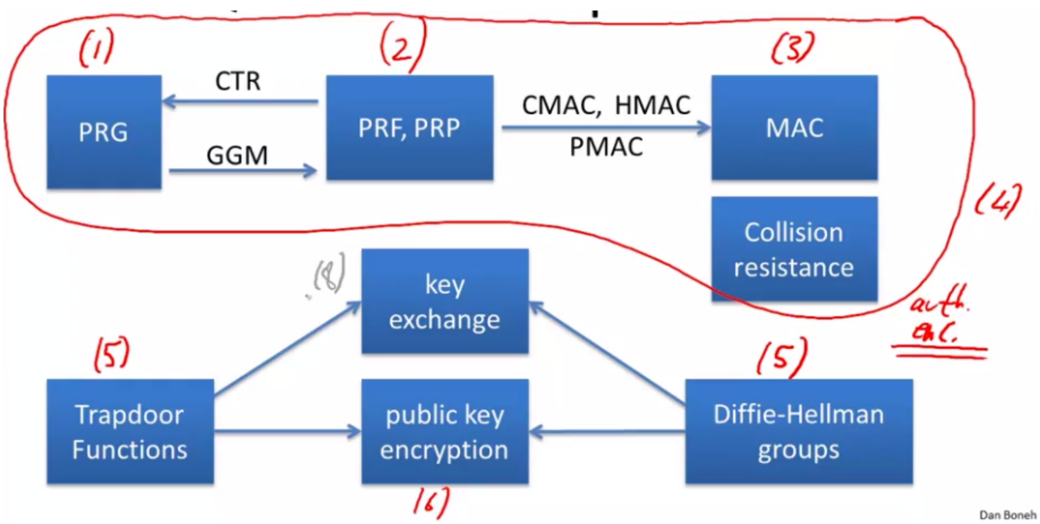
- choose random primes p,q ≈1024 bits. Set **N=pq**.
- choose integers **e , d** s.t. $\mathbf{e \cdot d = 1 \pmod{\varphi(N)}}$

Define: $f: \mathbb{Z}_N^* \rightarrow \mathbb{Z}_N^*$ as $\boxed{f(x) = x^e \text{ in } \mathbb{Z}_N}$

Lemma: f is one-way under the RSA assumption

Properties: $f(x \cdot y) = f(x) \cdot f(y)$ and **f has a trapdoor**

- Summary
    - Public key encryption
        - made possible by one way functions with special properties
        - homomorphic properties and trapdoors
            - F(x), F(y) => F(x + y) or F(x * y)
- **Farewell (For Now)**
- Quick review: primitives



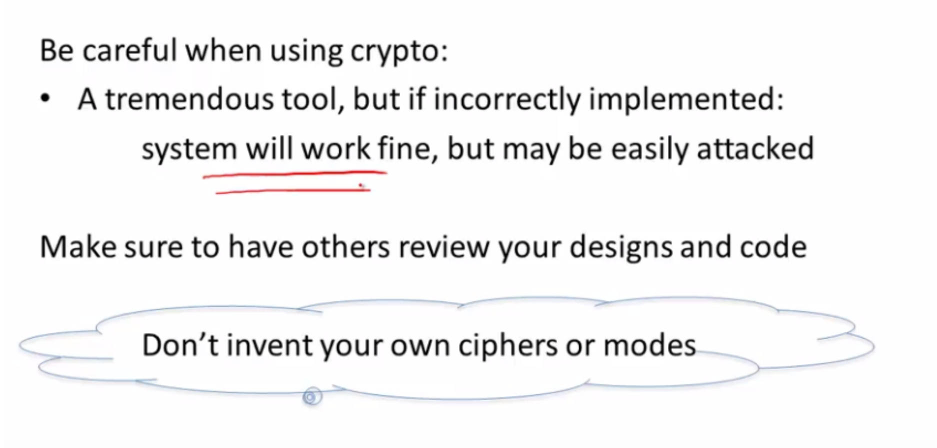
- Remaining core topics (part 2)

    - Digital signatures and certificates
    - Authenticated key exchange
    - User authentication:
      passwords, one-time passwords, challenge-response
    - Privacy mechanisms
    - Zero-knowledge protocols

- Man more topics to cover

    - Elliptic Curve Crypto
    - Quantum computing
    - New key management paradigms:
      identity based encryption and functional encryption
    - Anonymous digital cash
    - Private voting and auction systems
    - Computing on ciphertexts: fully homomorphic encryption
    - Lattice-based crypto
    - Two party and multi-party computation

- Final words


Be careful when using crypto:
• A tremendous tool, but if incorrectly implemented:
system will work fine, but may be easily attacked
Make sure to have others review your designs and code
Don’t invent your own ciphers or modes

-