Recap:

$A \xrightarrow{[m]} B$

(1) Bob: KeyGen: $pk = g^{sk}$ $\quad (y = g^{x})$

y: public key; x: secret key.

(2) Alice: Enc:

$$c = \langle g^r, m \cdot y^r \rangle$$

(3) Bob: Dec $\hookrightarrow y^r = (g^r)^x$

$\hookrightarrow (m \cdot y^r) \cdot y^{q-r} = m$

Zero-knowledge Proofs (ZkPs)

Given $\langle g, y = g^x \rangle$, prove you know x without revealing any information about it.

↳ prove you know a private key for a given public key

Attempt 1

Prover sends x'

Verifier check $g^{x'} \stackrel{?}{=} y$

↳ { Soundness: ✓
Zero knowledge: ✗

Attempt 2

Prover sends $x' + a = d$ ← random integer.

Verifier ⇝ ?

- Soundness: ×
- Zero knowledge: ✓

Attempt 3:

Prover chooses $a \in Z_q$.

Prover sends $d = x + a$

Prover sends $b = g^a$ ⇝ hides a under DLP.

$y = g^x$

Verifier $b \cdot y \stackrel{?}{=} g^d$

$g^a \cdot y^x \stackrel{?}{=} g^{x+a}$

$g^{a+x} \stackrel{?}{=} g^{x+a}$

- Soundness: ? ×
- Zero knowledge: ✓

↳ Can you produce b, d such that $b \cdot y = g^d$ without knowing x?

↳ Attempt: choose b
↳ $d = \log_g(by)$
↳ × (DLP)

↳ Attempt: choose d)
↳ $b = y^{-1} g^d = g^{d-x}$
↳ ✓

↳ Not a sound proof.

Attempt 4:

Do Attempt 3 and then send value a (Attempt 1' for $b = g^a$)

↳ { sound ✓ (can't fake b, d and reveal a s.t. $g^a = b$)
ZK ? ×

$View_{Adv} = \langle d = a + x, b = y^a, a \rangle$
↳ $x = d - a$
✓ ✓ ✓

Attempt 5:

* Start with Attempt 3:

  Prover chooses $a \in_r Z_q$

  Prover sends $b = g^a$

* Then Verifier will flip a coin
  - ↳ if heads: Prover uses attempt 1 on $b \rightarrow$ sends a
  - ↳ if tails: Prover sends $d = x + a$

* Verifier will check
  - ↳ if heads: $b \stackrel{?}{=} g^a$
  - ↳ if tails: $b \cdot y \stackrel{?}{=} g^d$

* { Soundness: ✓
  { Zero knowledge: ✓

Zero knowledge:

Heads → not sending anything related to x

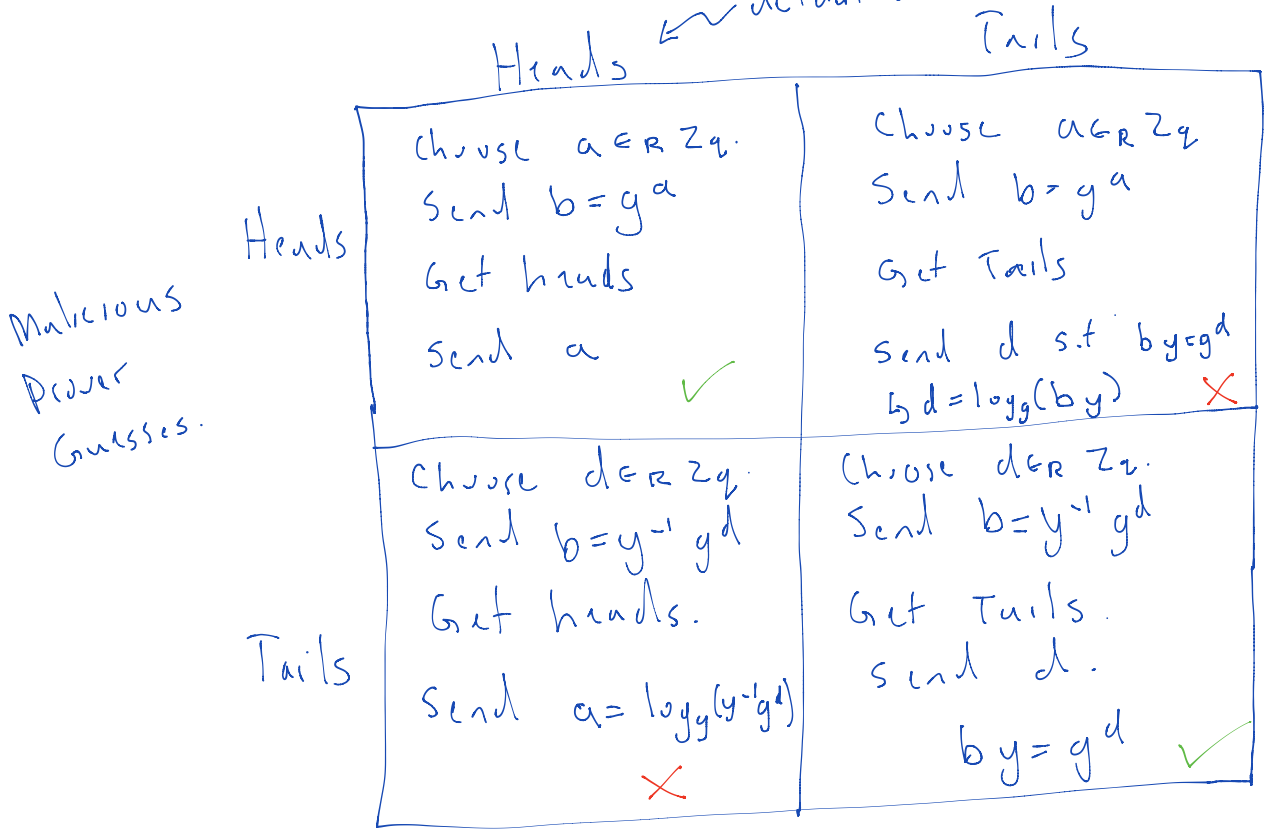
Tails → send $d = x + a$ and $b = g^a$

② x is unknown b/c a is | ① a is unknown b/c of DLP

Soundness:

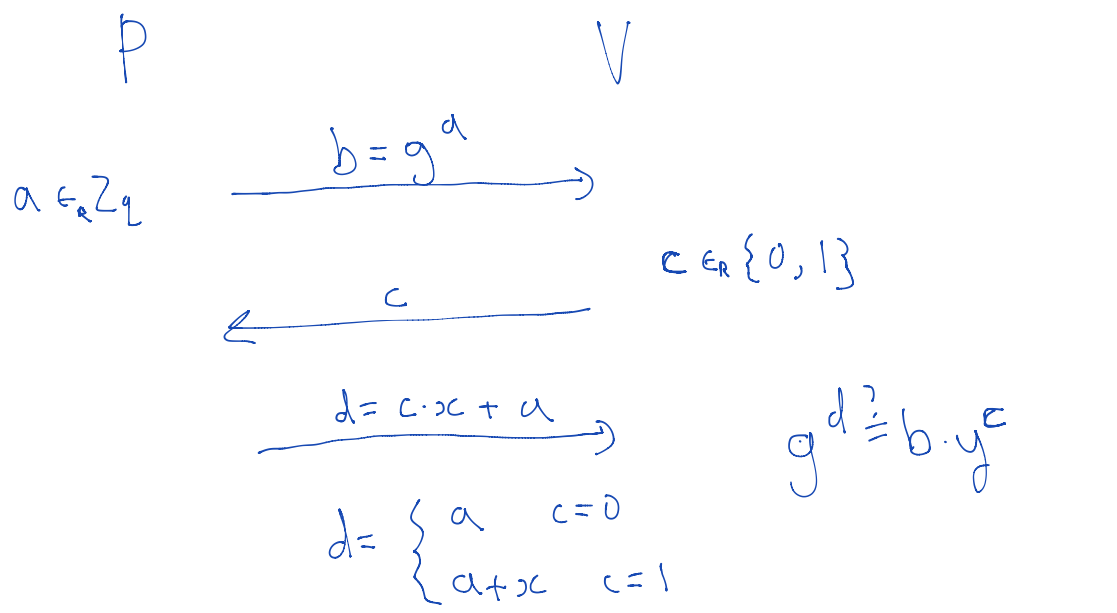
* If Prover knows x, they can always answer (completeness).

* If the prover doesn't x, they will be unable to give the verifier correct $\langle b, d \rangle$ pair with probability $\frac{1}{2}$.



Conclusion: it is sound 50% of the time.

Rewrite with slightly different notation:



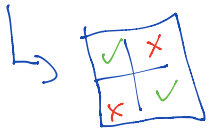
Variation 1:

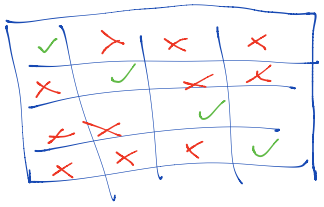
Choose larger challenge $c \in_R \{0,1\}^t$

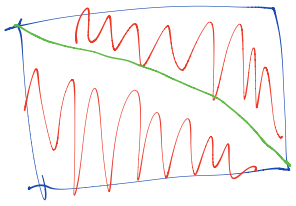
↳ same as doing the protocol t times in terms of soundness

↳ $Pr[\text{cheating}] = 1/2^t$

↳ $t=20 \rightarrow 1 - Pr[\text{cheating}] < 99.9999\%$

↳ 





Variation 2

We want a non-interactive version where prover can choose c by themselves

↳ however if prover chooses c, they can cheat

↳ interactive: verifier chooses c after prover sends b.

↳ P —b→ V
V —c→ P
P —d→ V

⇒

P —b→ V
P —c=H(b)→ V
P —d→ V

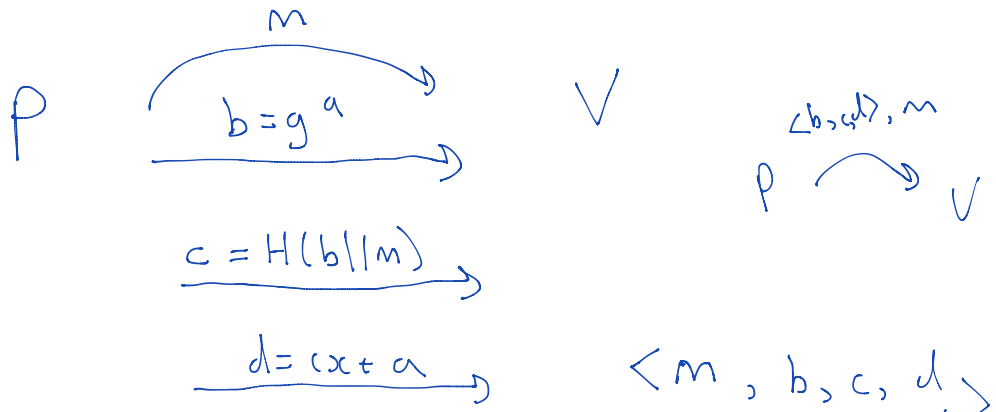
If c=H(b), P must have chosen b before choosing c (PR of hash)

↳ therefore it is sound.

↳ Fiat-Shamir Heuristic

Variation 3:

* "Staple" a message to a ZKP by including it in the hash that generates the challenge.
  - ↳ this acts as a signature on the message by the person who knows the secret key of a given public key.



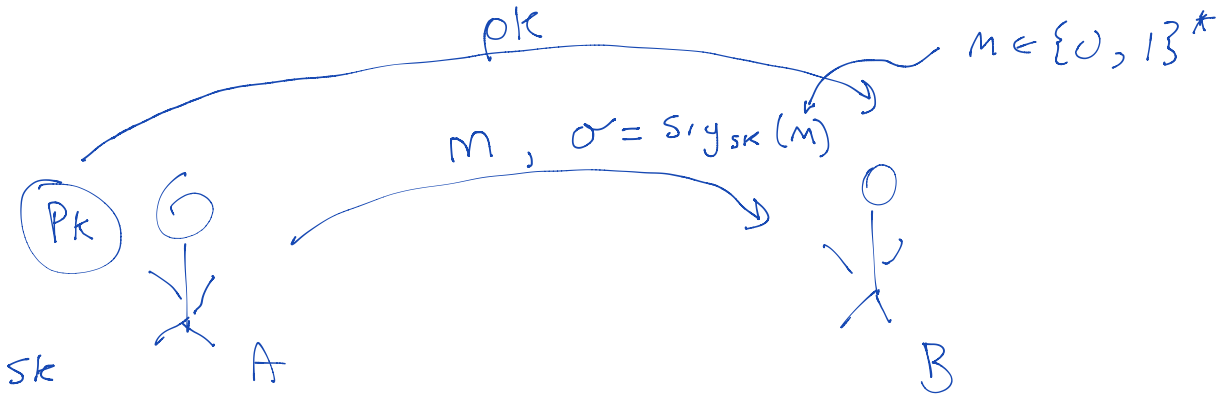
Verification:

$c = H(b, m)$

$b \cdot y^c \stackrel{?}{=} g^d$

Digital Signature

(Same as above in different order).



(1) Key Generation

$sk = x$

$pk = y = g^x$

(2) Signing

$\sigma = Sig_{sk}(m, r)$

$= \langle b, c, d \rangle$

$= \langle g^r, H(b, m), r + cx \rangle$

(3) Verification.

$T/F = Verify_{pk}(m, \sigma)$

$$T = \begin{cases} c \stackrel{?}{=} H(b, m) \\ g^d \stackrel{?}{=} b \cdot y^c \end{cases}$$ pk

DSA:

* Above is "Schnorr" signature
* DSA is more common.

	Schnorr	DSA
$d =$	$a + cx$	$a^{-1}(c + bx)$

Implementation Notes.

(1) Both Schnorr & DSA: use the value a (randomness) to sign two diff messages → compute the private key.
↳ Homework.

(2) The hash output should be $|c| \geq 112$ bits so probability of hash being a guessed value is $1/2^{112} \sim 2^{112}$ guesses.
↳ infeasible.

However we also want collision resistance

↳ if $H(b, m_1) = H(b, m_2)$ for different $m_1 \neq m_2$

↳ $sig(m_1) = sig(m_2)$

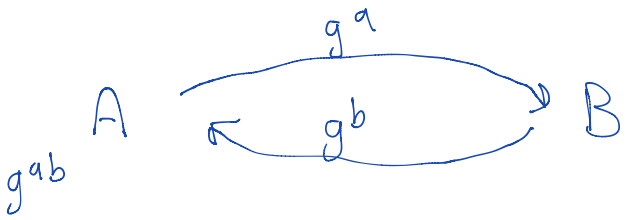
↳ $|c| \geq 224$-bits.

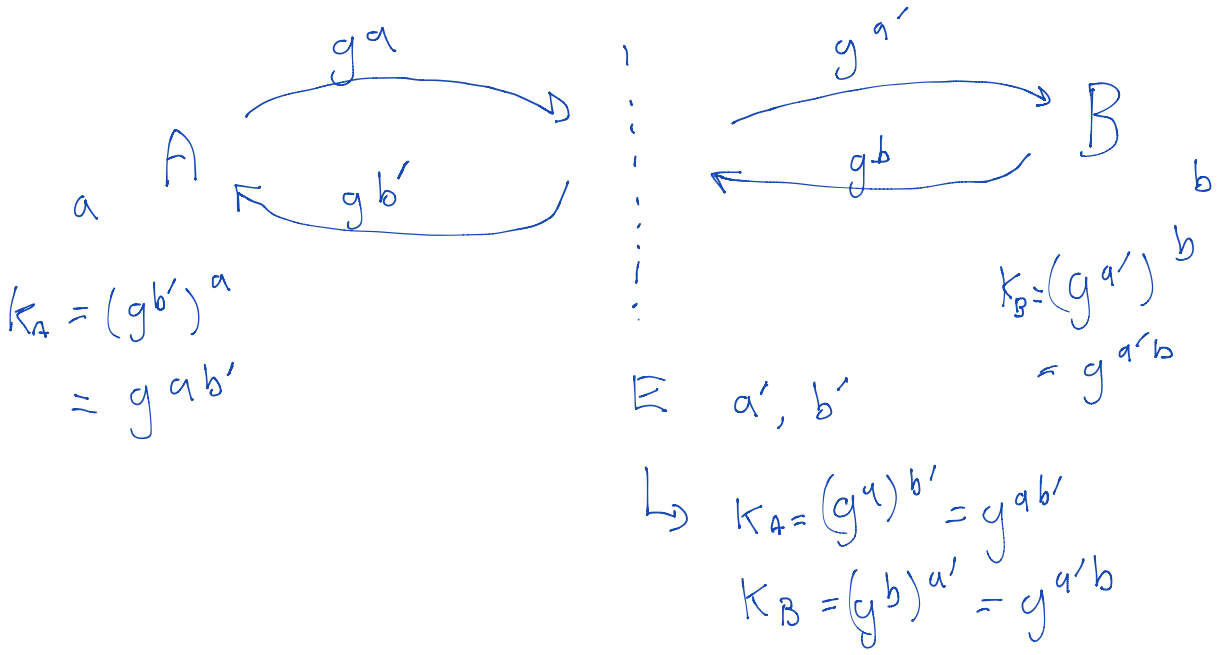
(3) Because messages are hashed, you can sign messages of any length.

↳ "hash and sign" paradigm

Diffie-Hellman Revisited

Earlier we noted DHE was only secure against a passive adversary.

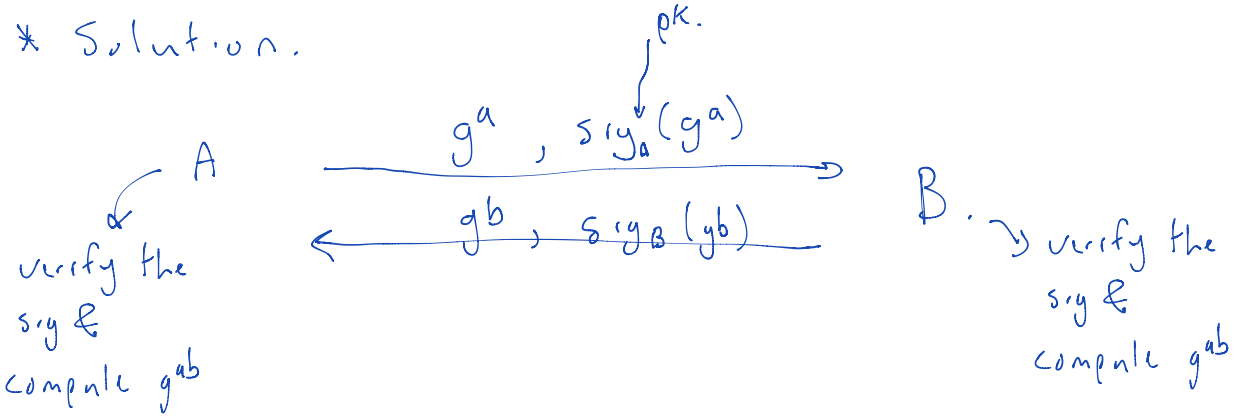




A <-DH-> E <-DH-> B

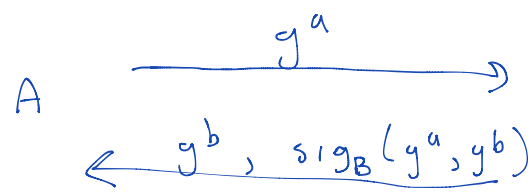
* Eve can decrypt in the middle and re-encrypt.

* Solution.



"Station-to-Station" Protocol

* Simpler Variant



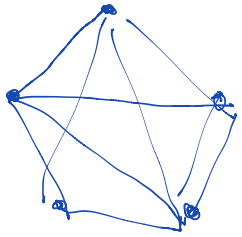
Alice will abort if g^a that Bob signed doesn't match what she sent.

One last problem:

How does Alice know what Bob's public key is?

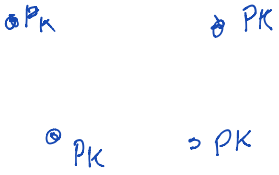
↳ PKI (Public key infrastructure)

Symmetric key.



$\binom{n}{2}$ keys $\simeq n^2$

Public key.



n keys.

Certificate Authority.

$sig_{CA}(pk)$ • $sig_{CA}(pk) \rightarrow$ certificate.

• $sig_{CA}(pk)$

$sig_{CA}(pk)$ • $sig_{CA}(pk)$

have to know

↓

1 key