

COS433/Math 473: Cryptography

Mark Zhandry

Princeton University

Fall 2020

Announcements/Reminders

HW5 due TODAY

HW6 released soon

PR2 due Dec 5

Previously on COS 433...

Identification Protocols

Identification



Identification



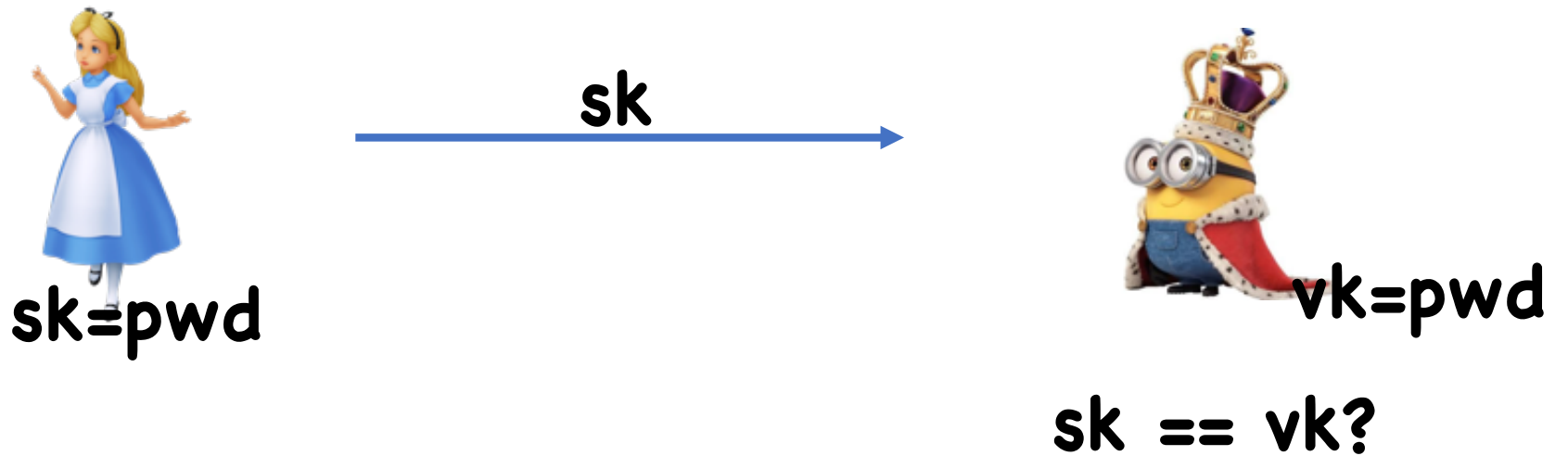
Types of Attacks

Direct Attack:



Basic Password Protocol

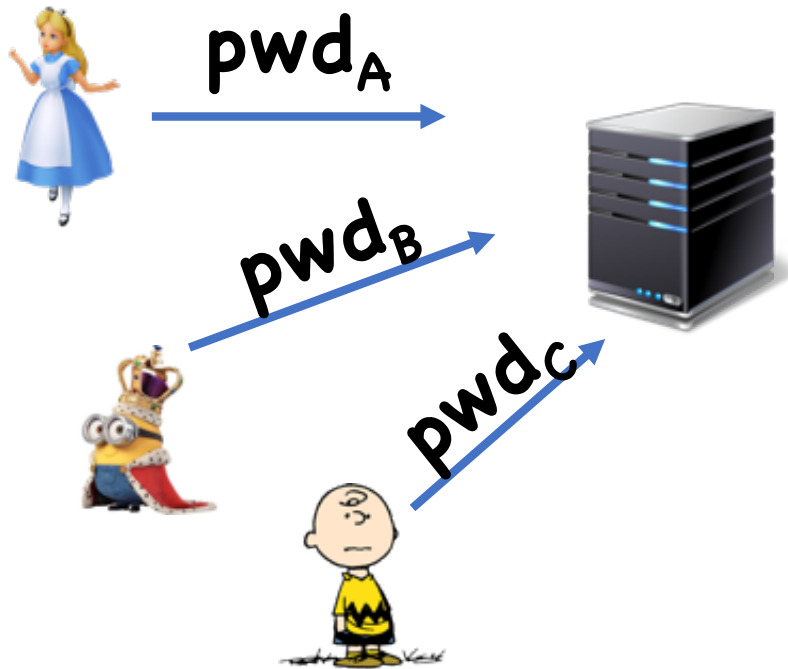
Never ever (ever ever...) use



Salting

Let H be a hash function

s_i random



User	Salt	Pwd
Alice	s_A	$H(s_A, pwd_A)$
Bob	s_B	$H(s_B, pwd_B)$
Charlie	s_C	$H(s_C, pwd_C)$
...

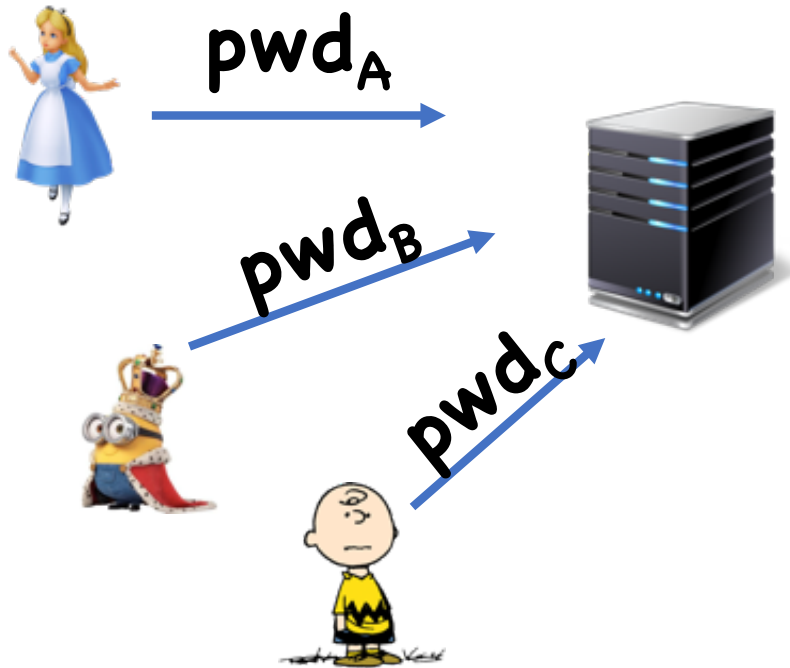
Today

Identification continued

Zero knowledge

Crypto from minimal assumptions (if time)

Encrypt Passwords?



User	Pwd
Alice	$Enc(k, pwd_A)$
Bob	$Enc(k, pwd_B)$
Charlie	$Enc(k, pwd_C)$
...	...

Encrypt Passwords?

Again, never ever (ever ever....) use

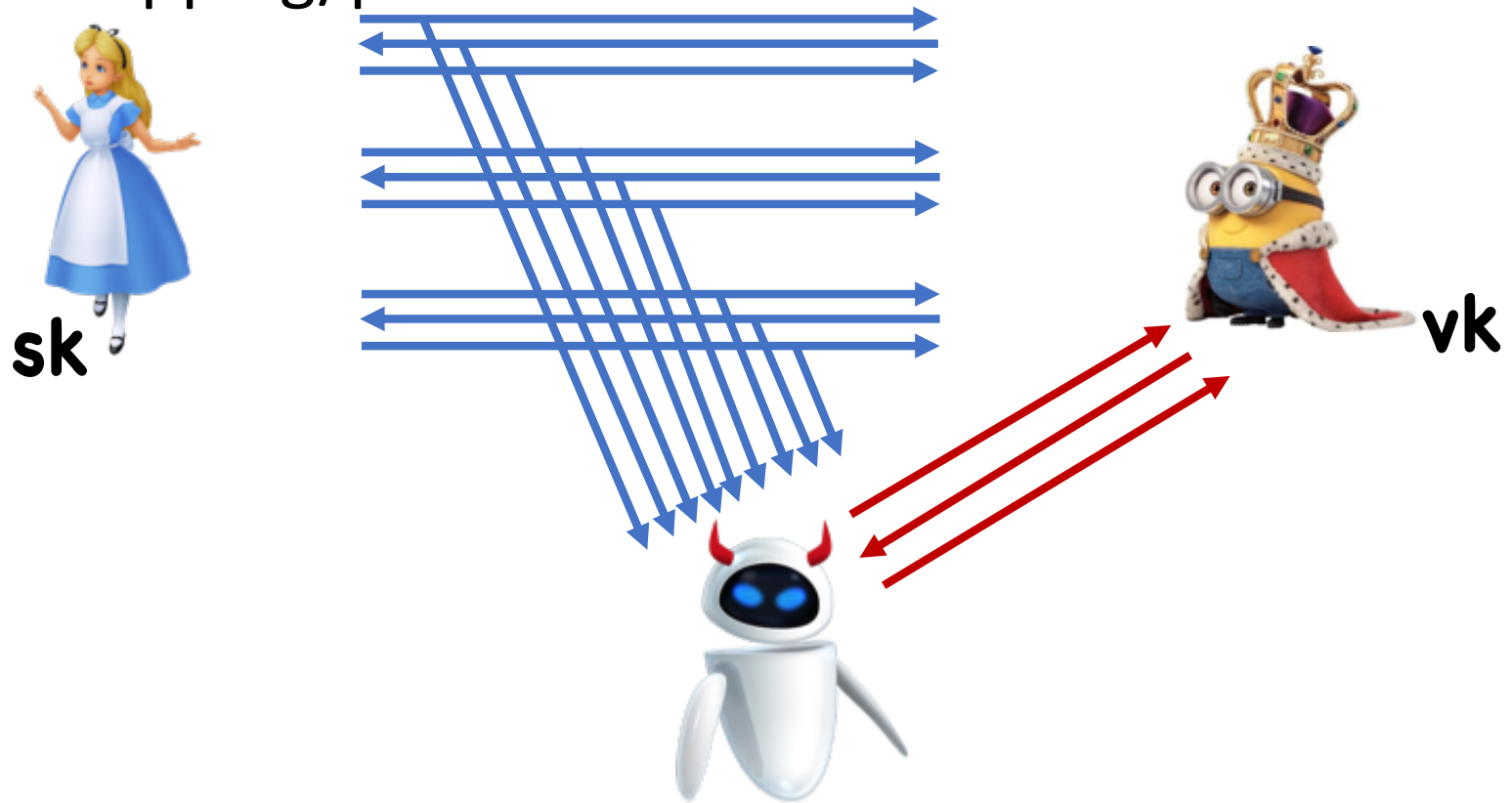
- To check password, need to decrypt
- Must store decryption key **k** somewhere
- What if **k** is stolen?

Need to use one-way mechanism

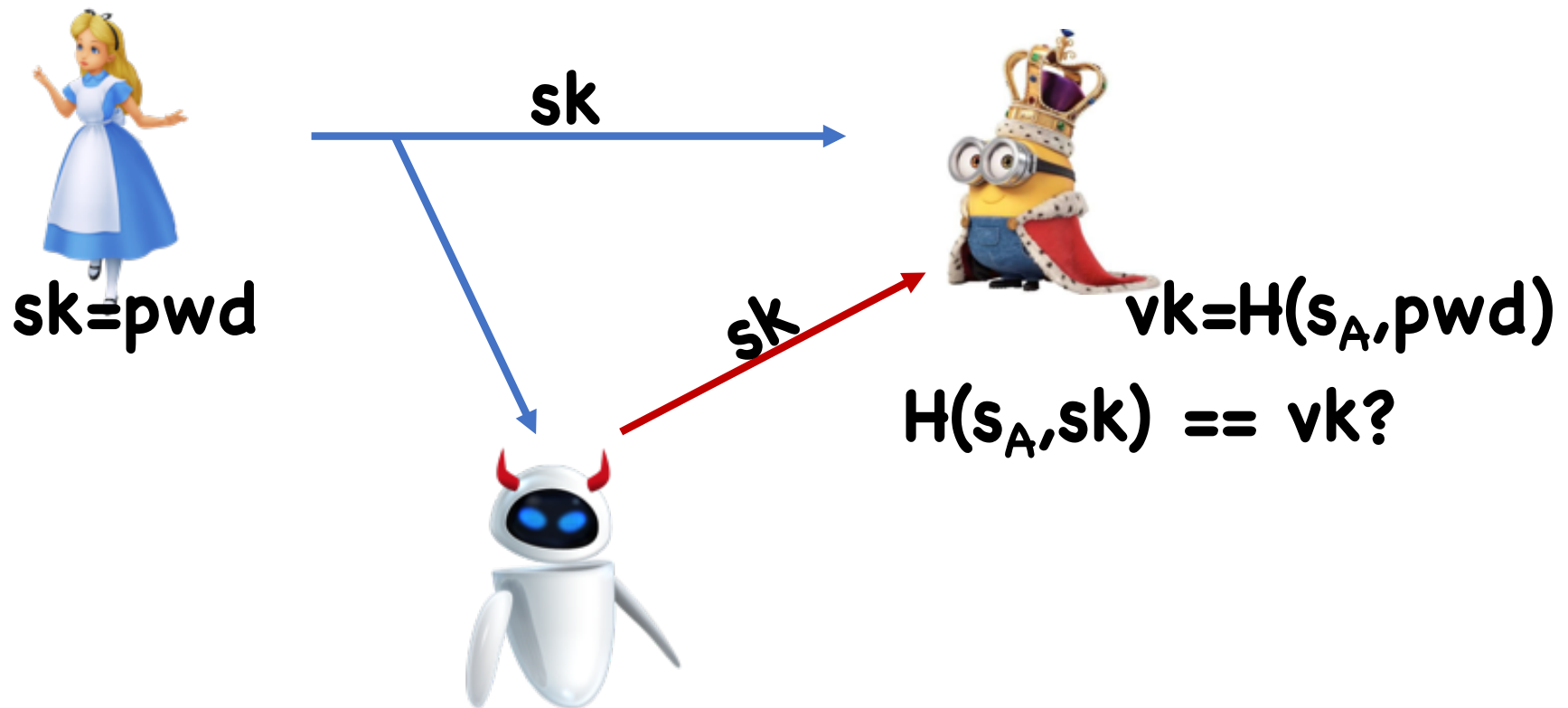
- With hash function, not even server can recover password

Types of Attacks

Eavesdropping/passive:



Security Against Eavesdropping



Security Against Eavesdropping

One solution: update **sk,vk** after every run

One-time Passwords

Let \mathbf{F} be a PRF



$sk=(k,0)$

$$sk_0 = F(k,0)$$



$vk=(k,0)$

$sk_0 == F(k,0)?$

One-time Passwords

Let \mathbf{F} be a PRF



$$sk_1 = F(k,1)$$

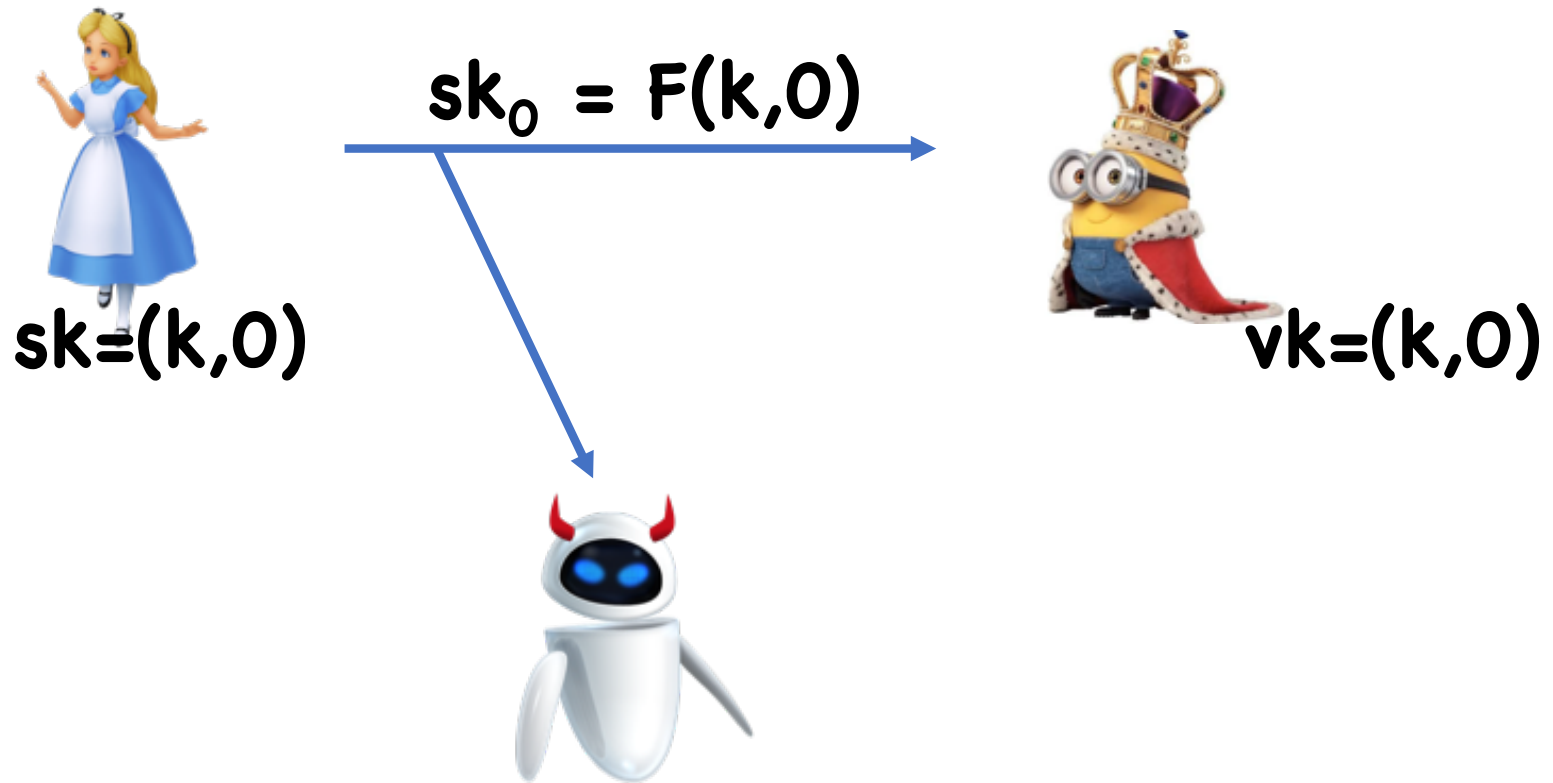


$$vk=(k,1)$$

$$sk_1 == F(k,1)?$$

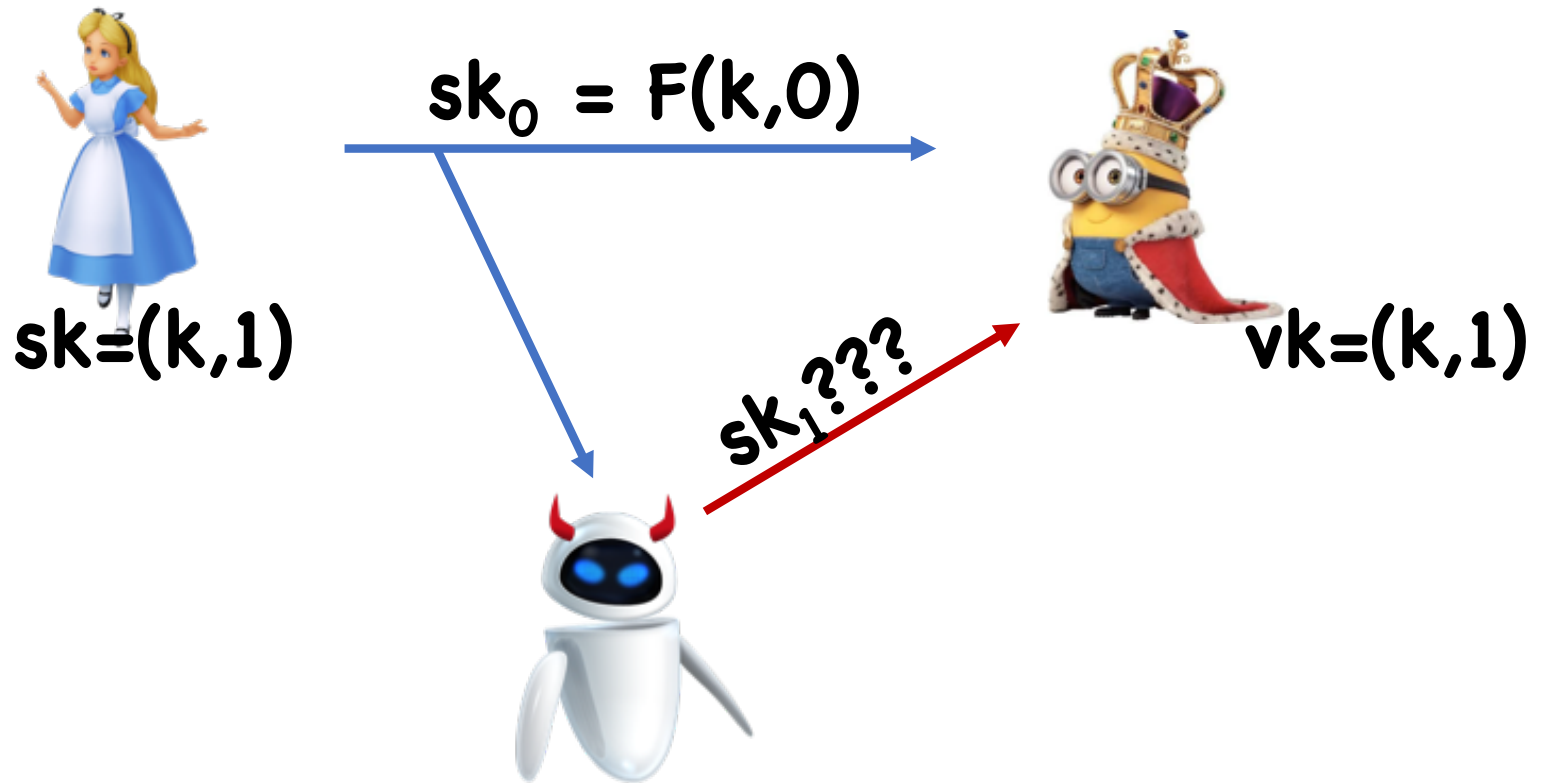
One-time Passwords

Let \mathbf{F} be a PRF



One-time Passwords

Let \mathbf{F} be a PRF



One-time Passwords

Advancing state:

- Time based (e.g. every minute, day, etc)
- User Action (button press)

Must allow for small variation in counter value

- Clocks may be off, user may accidentally press button



Stateless Schemes?

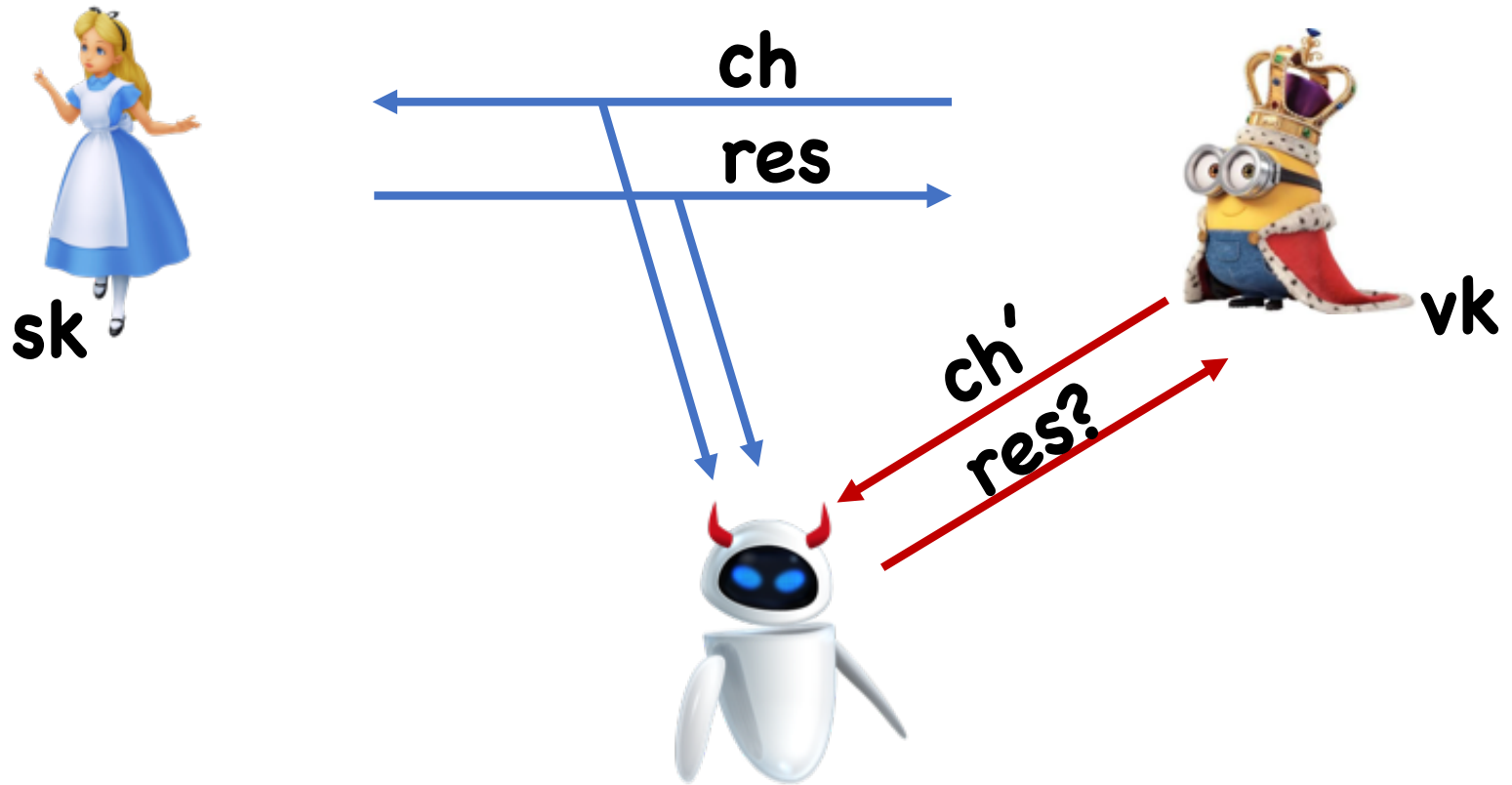
So far, all schemes secure against eavesdropping are stateful

Easy theorem: any one-message stateless ID protocol is insecure if the adversary can eavesdrop

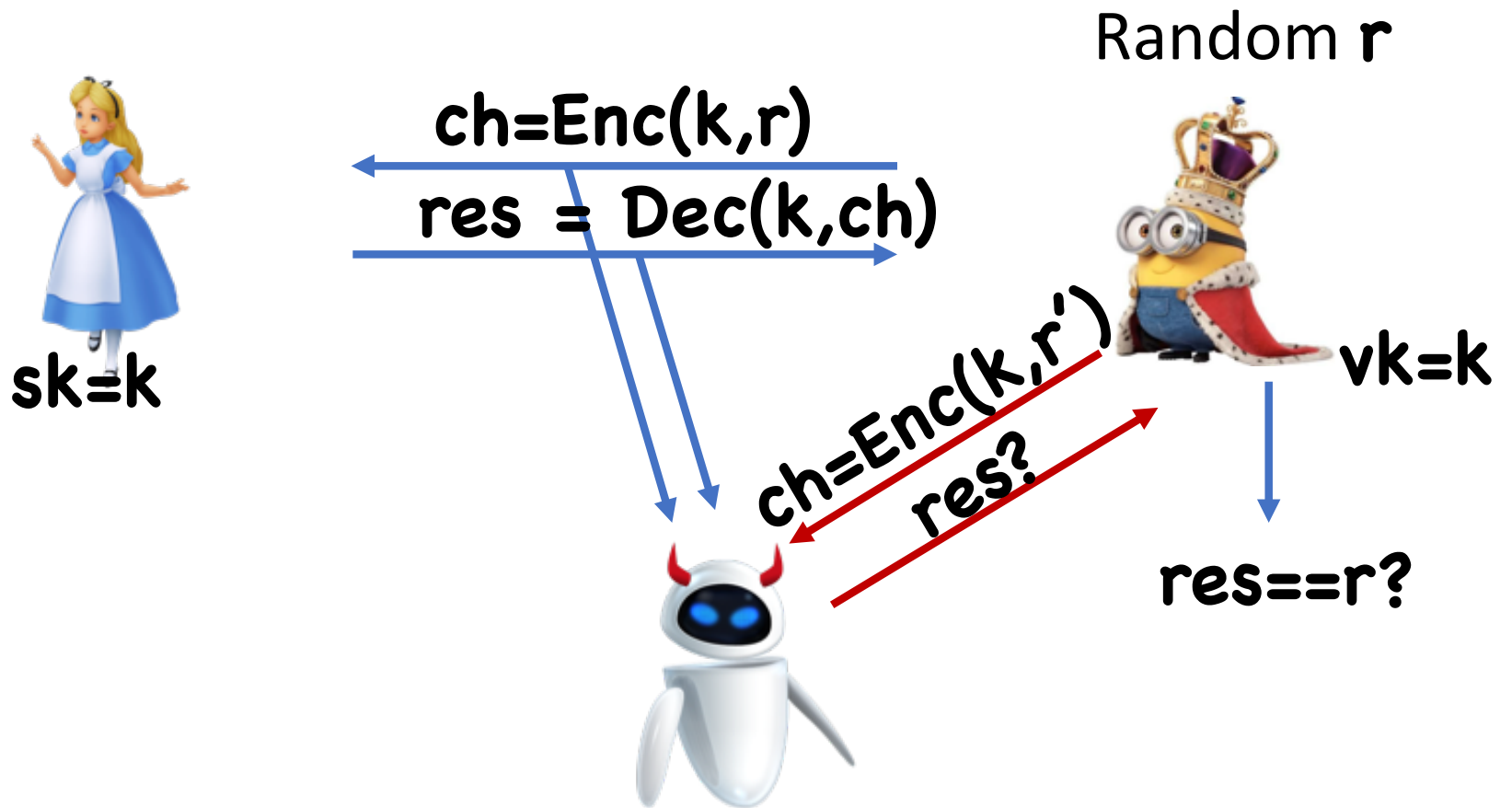
- Simply replay message

If want stateless scheme, instead want at least two messages

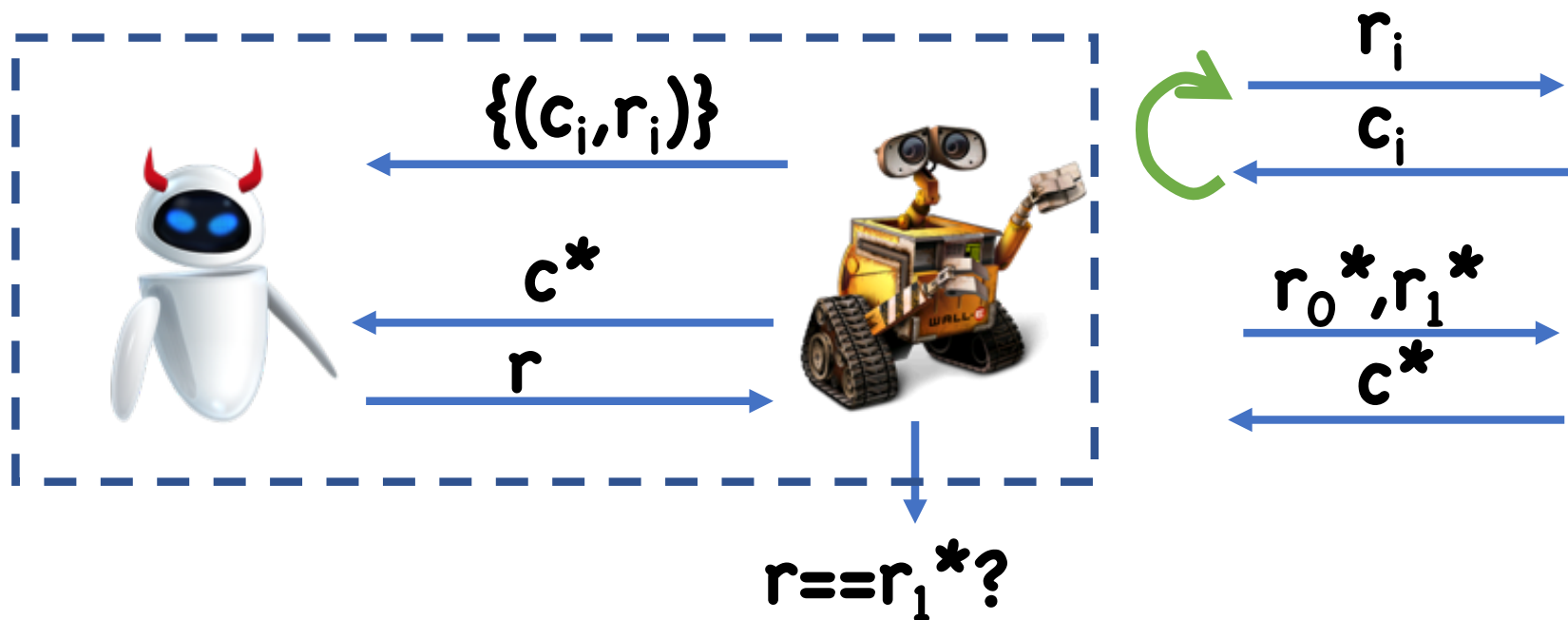
Challenge-Response



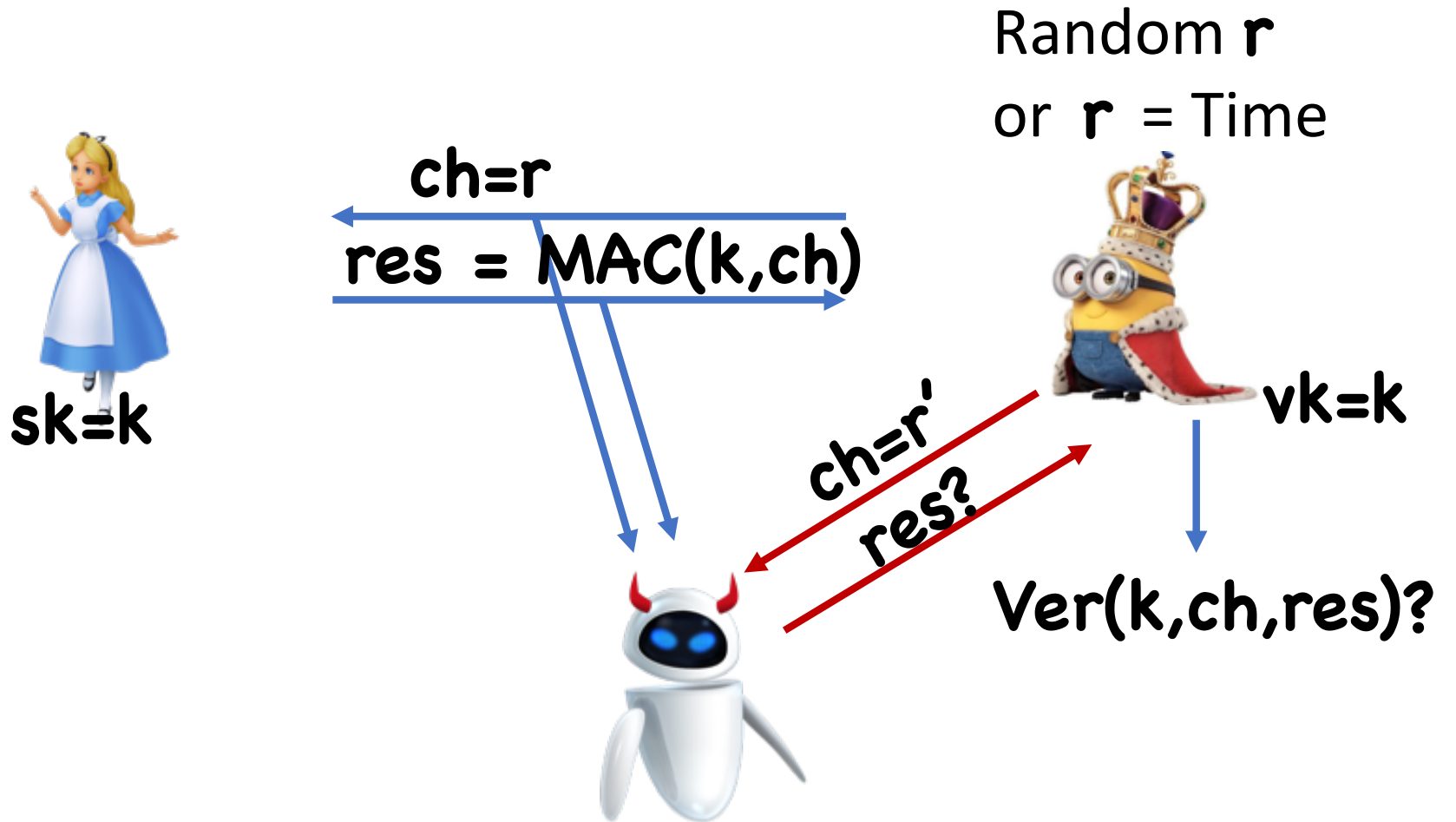
C-R Using Encryption



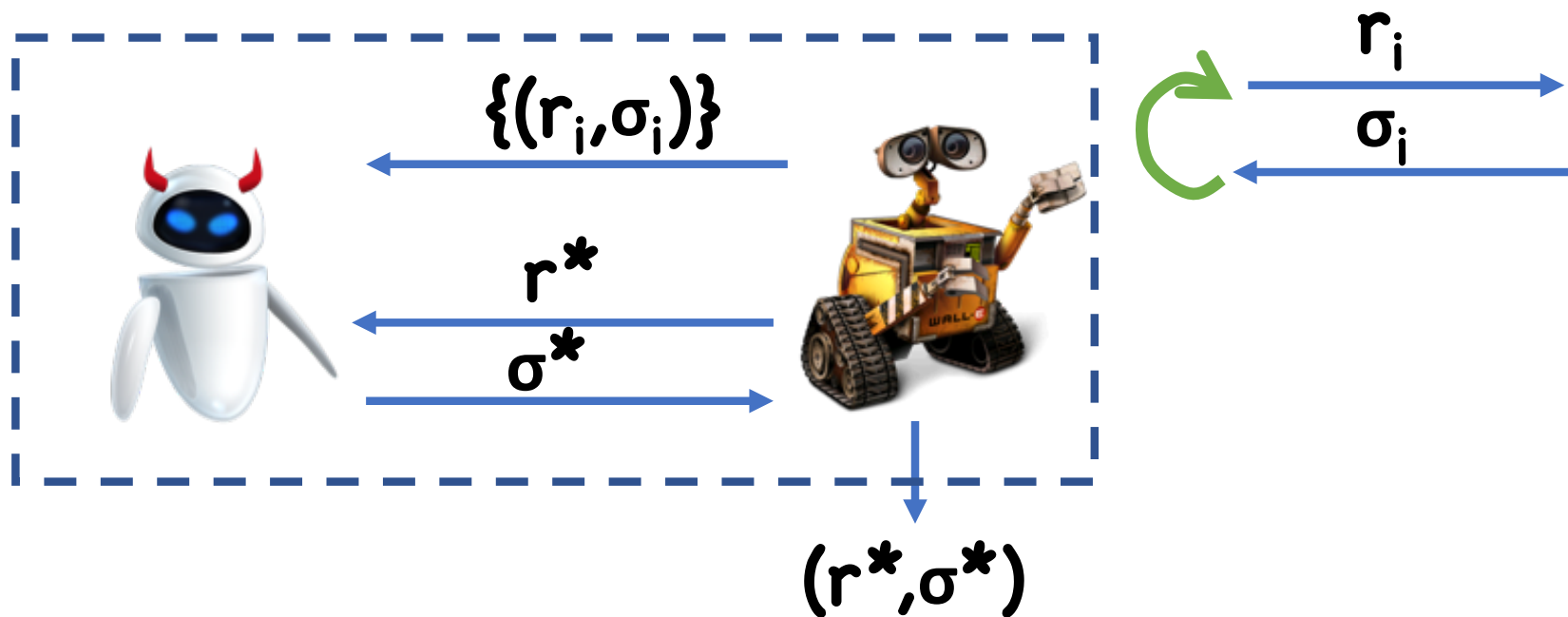
Theorem: If (Enc, Dec) is a CPA-secure secure SKE/PKE scheme, then the C-R protocol is a secret key/public key identification protocol secure against eavesdropping attacks



C-R Using MACs/Signatures

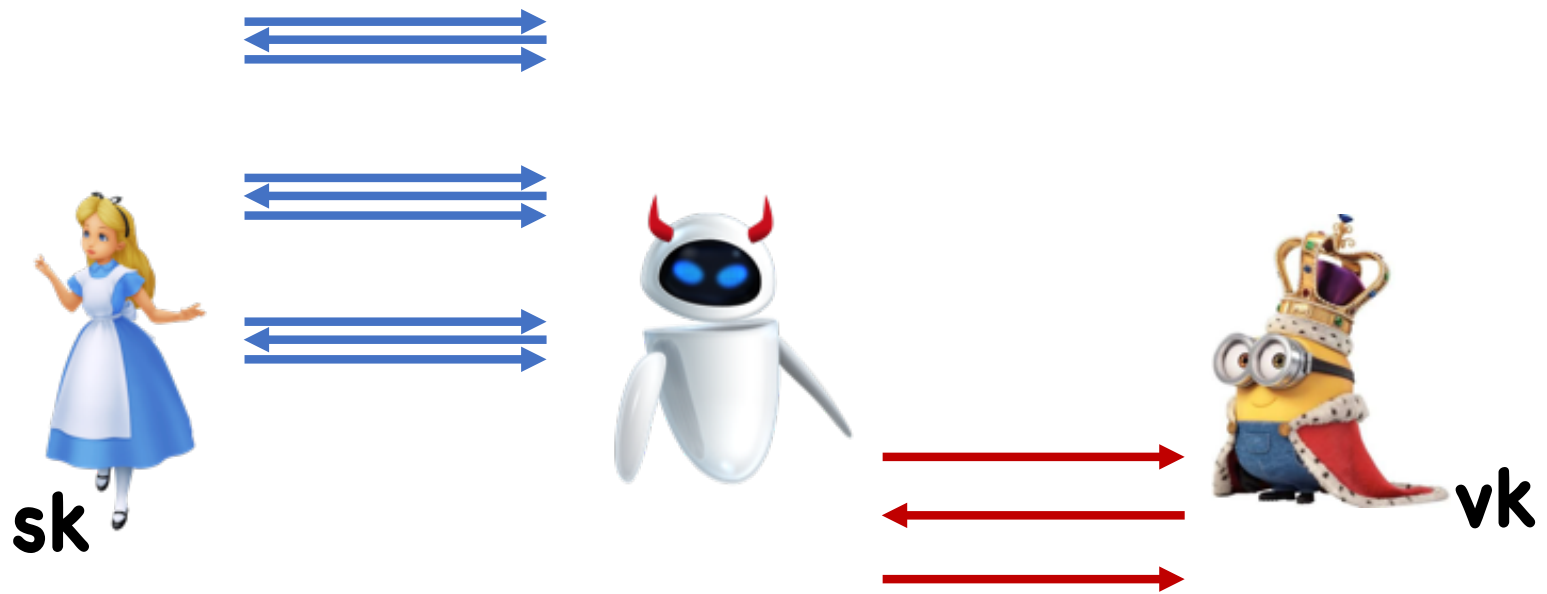


Theorem: If **(MAC, Ver)** is a CMA-secure secure MAC/Signature scheme, then the C-R protocol is a secret key/public key identification protocol secure against eavesdropping attacks



Types of Attacks

Man-in-the-Middle/Active:



Active Attacks

For enc-based C-R, CPA-secure is insufficient

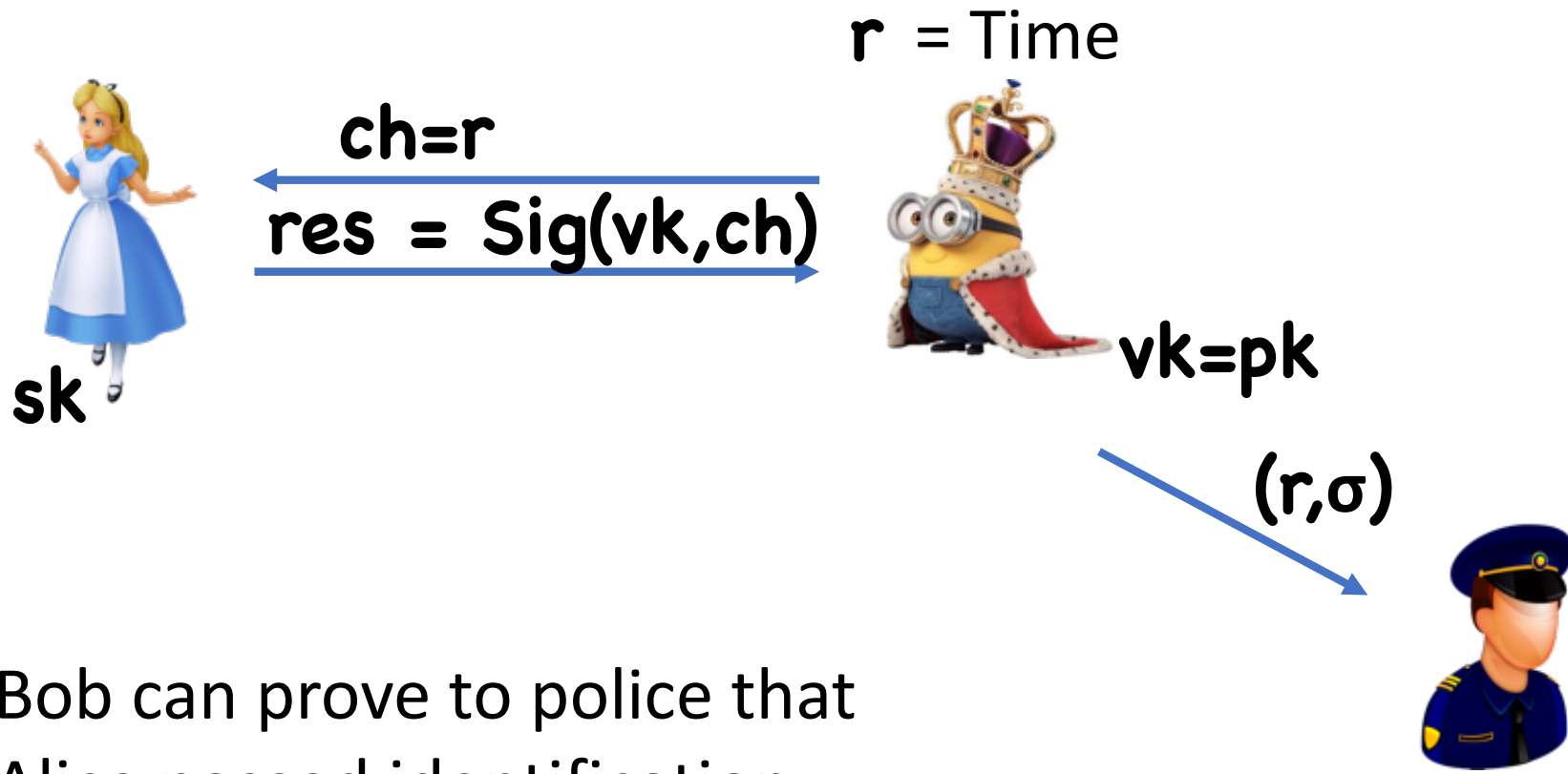
- Instead need CCA-security (lunch-time sufficient)

For MAC/Sig-based C-R, CMA-security is sufficient

Zero Knowledge

Non-Repudiation

Consider signature-based C-R



Zero Knowledge

What if Bob could have come up with a valid transcript, without ever interacting with Alice?

- Then Bob cannot prove to police that Alice authenticated

Seems impossible:

- If (public) **vk** is sufficient to come up with valid transcript, why can't an adversary do the same?

Zero Knowledge

Adversary CAN come up with valid transcripts, but Bob doesn't accept transcripts

- Instead, accepts *interactions*

Ex: public key Enc-based C-R

- Valid transcript: (\mathbf{c}, \mathbf{r}) where \mathbf{c} encrypts \mathbf{r}
- Anyone can come up with a valid transcript
- However, only Alice can generate the transcript for a given \mathbf{c}

Takeaway: order of messages matters

Zero Knowledge Proofs

Mathematical Proof

Statement x

Witness/proof w



w



$Ver(x,w)$

Interactive Proof

Statement x

Witness w



Properties of Interactive Proofs

Let (P, V) be a pair of probabilistic interactive algorithms for the proof system

Completeness: If w is a valid witness for x , then V should always accept

Soundness: If x is false, then no cheating prover can cause V to accept

- Perfect: accept with probability 0
- Statistical: accept with negligible probability
- Computational: cheating prover is comp. bounded

Zero Knowledge


Intuition: verifier doesn't learn anything by engaging in the protocol (other than the truthfulness of x)

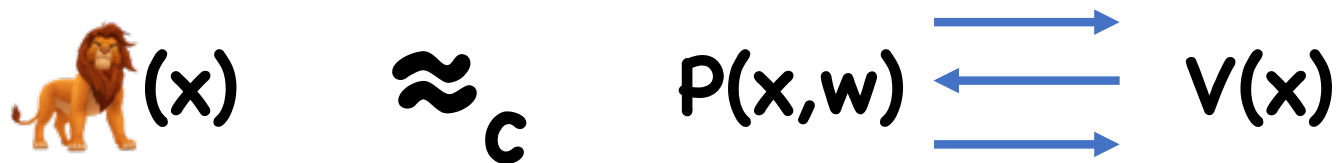
How to characterize what adversary “knows”?

- Only outputs a bit
- May “know” witness, but hidden inside the program's state

Zero Knowledge

First Attempt:

\exists “simulator”  s.t. for every true statement \mathbf{x} ,
valid witness \mathbf{w} ,



Zero Knowledge

First Attempt:

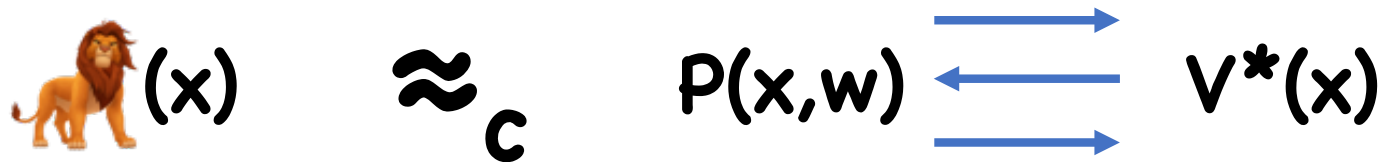
Assumes Bob obeys protocol

- “Honest Verifier”

But what if Bob deviates from specified prover algorithm to try and learn more about the witness?

Zero Knowledge

For every malicious verifier V^* , \exists “simulator”  s.t. for every true statement x , valid witness w ,



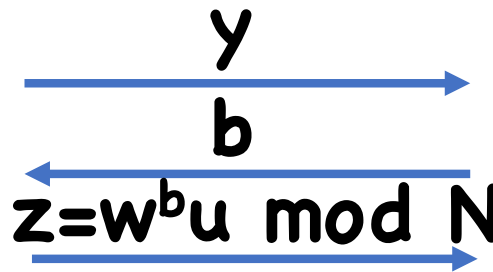
QR Protocol

Statements: x is a Q.R. mod N

Witness: w s.t. $w^2 \bmod N = x$

Protocol:

$u \leftarrow \mathbb{Z}_N^*$
 $y \leftarrow u^2 \bmod N$



$b \leftarrow \{0,1\}$

$z^2 \stackrel{?}{=} x^b y \bmod N$

QR Protocol

Zero Knowledge:

What does Bob see?

- A random QR \mathbf{y} ,
- A random bit \mathbf{b} ,
- A random root of $\mathbf{x}^{\mathbf{b}}\mathbf{y}$

Idea: simulator chooses \mathbf{b} , then \mathbf{y} ,

- Can choose \mathbf{y} s.t. it always knows a square root of $\mathbf{x}^{\mathbf{b}}\mathbf{y}$

QR Protocol

Honest Verifier Zero Knowledge:



(x):

- Choose a random bit **b**
- Choose a random string **z**
- Let $y = x^{-b}z^2$
- Output **(y,b,z)**

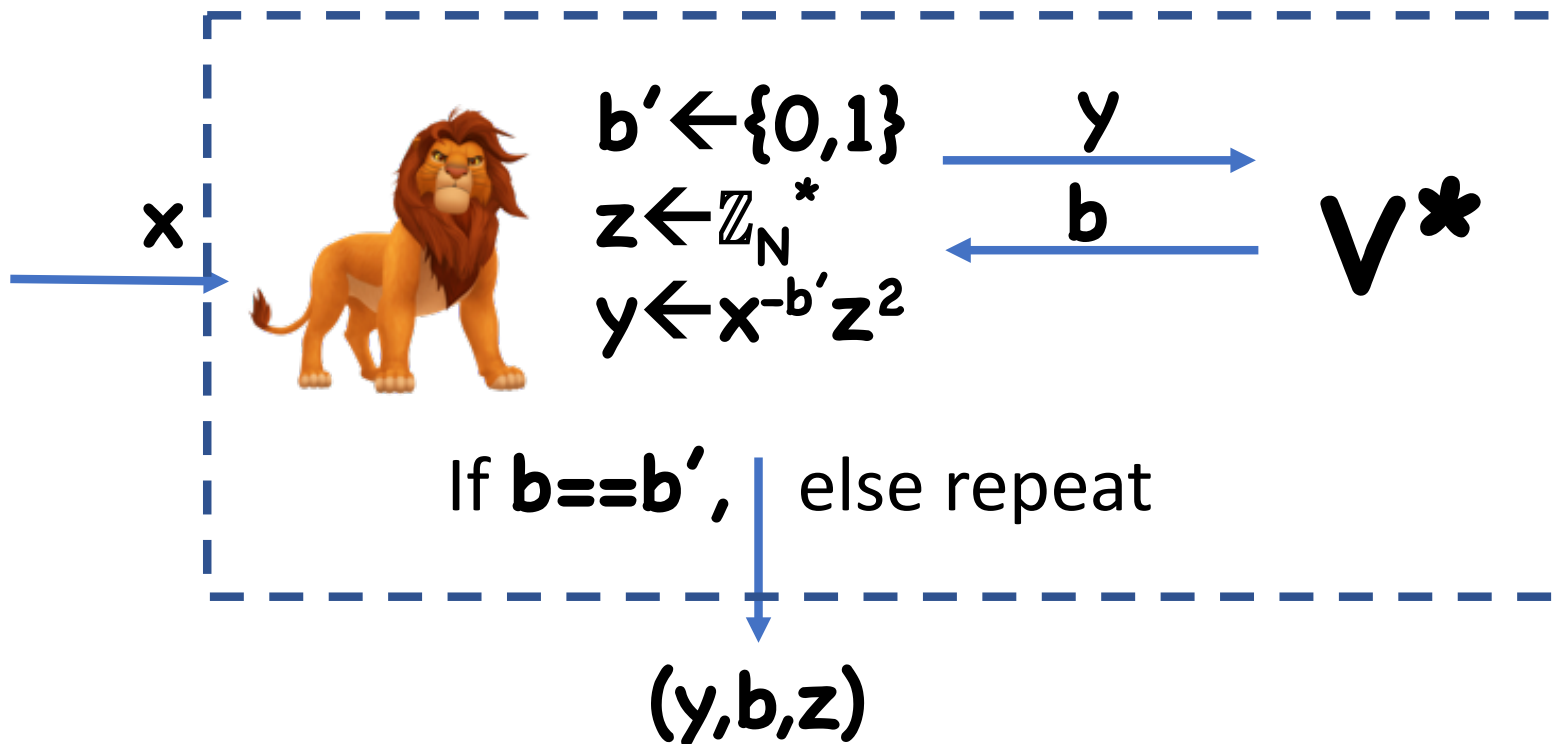
- If **x** is a QR, then **y** is a random QR, no matter what **b** is
- **z** is a square root of $x^b y$



(y,b,z) is distributed identically to **(P,V)(x)**

QR Protocol

(Malicious Verifier) Zero Knowledge:



QR Protocol

(Malicious Verifier) Zero Knowledge:


Proof:

- If \mathbf{x} is a QR, then \mathbf{y} is a random QR, independent of \mathbf{b}'
- Conditioned on $\mathbf{b}' = \mathbf{b}$, then $(\mathbf{y}, \mathbf{b}, \mathbf{z})$ is identical to random transcript seen by \mathbf{V}^*
- $\mathbf{b}' = \mathbf{b}$ with probability $1/2$

Repetition and Zero Knowledge

(sequential) repetition also preserves ZK

Unfortunately, parallel repetition might not:

-  makes guesses $\mathbf{b}_1', \mathbf{b}_2', \dots$
- Generates valid transcript only if all guesses were correct
- Probability of correct guess: 2^{-t}

Maybe other simulators will work?

- Known to be impossible in general, but nothing known for QR

Zero Knowledge Proofs

Known:

- Proofs for any NP statement assuming statistically-binding commitments
- Non-interactive ZK proofs for any NP statement using trapdoor permutations