CS 528 (Fall 2021) Data Privacy & Security

Yuan Hong
Department of Computer Science
Illinois Institute of Technology

Chapter 9-A
Zero-Knowledge Proof

OUTLINE

- Zero Knowledge Proof
- Cryptographic Commitment

INTERACTIVE PROOF SYSTEMS

Traditionally, a proof for a statement is a static string such that one can verify for its correctness

- Follows axioms and deduction rules.

Generalizing proof systems to be interactive

- A proof system involves an algorithm for a prover and a verifier.
- A proof system can be probabilistic in ensuring correctness of the statement being proved
- The verifier accepts or rejects the proof after multiple challenges and responses

ZERO KNOWLEDGE PROOFS

A protocol involving a prover and a verifier that enables the prover to prove to a verifier without revealing any other information

- E.g., proving that a number n is of the form of the product of two prime numbers
- Proving that one knows p, q such that n=pq
- Proving that one knows x such $g^x \bmod p = y$

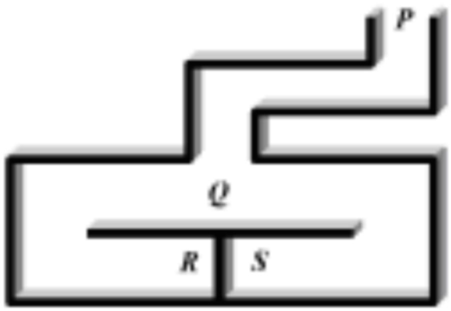
Computational Efficiency – No Encryption

- E.g., proving that a number n is of the form of the product of two prime number

CLASSIC EXAMPLE

- Ali Baba's Cave

  Alice has to convince Bob She knows the secret to open the cave door without telling the secret ("Open Sesame").



(source: http://www.rsasecurity.com/rsalabs/faq/2-1-8.html)

ANOTHER EXAMPLE

- Alice would like to prove that she has the key for a room to Bob.
  - (1) Alice shows the key to Bob
  - (2) Bob knows that there is a unique item in the room; Alice opens the door using her key; Alice shows the box to Bob

ANOTHER EXAMPLE

- In public key cryptosystem,
- Alice has Bob's public key, but Alice has never met Bob. One day, Bob recognizes Alice. Alice is unsure if the person is Bob or not. Bob has to prove that he is Bob to Alice.
  - Bob gives his private key to Alice. Alice tests it with the public key and any message
  - Alice generates a random value; encrypts it with Bob's public key, and sends it to Bob; Bob decrypts it and shows it to Alice;

TWO KINDS OF ZK PROOFS

ZK proof of a statement

- convincing the verifier that a statement is true without yielding any other information
- example of a statement, a propositional formula is satisfiable

ZK proof of knowledge

- convincing the verifier that one knows a secret, e.g., one knows the discrete logarithm $\log_g(y)$

PROPERTIES OF INTERACTIVE ZKP OF KNOWLEDGE

Completeness

- Given honest prover and honest verifier, the protocol succeeds with overwhelming probability

Soundness

- No one who doesn't know the secret can convince the verifier with nonnegligible probability

Zero knowledge

- The proof does not leak any additional information

FIAT-SHAMIR PROTOCOL FOR PROVING QUADRATIC RESIDUES

ILLINOIS INSTITUTE OF TECHNOLOGY

Statement: x is QR modulo n

Prover knows w such that w^2=x (mod n)

Repeat the following one-round protocol t times

One-round Protocol:

- P to V: y = r^2 mod n, where r is randomly chosen
- V to P: b $\leftarrow$ {0,1}, randomly chosen
- P to V: z=rw^b, i.e., z=r if b=0, z=rw if b=1
- V verifies: z^2=yx^b, i.e., z^2=y if b=0, z^2=yx if b=1

In number theory, an integer *q* is called a **quadratic residue** modulo *n* if it is congruent to a perfect square modulo *n*; i.e., if there exists an integer *x* such that:

$$x^2 \equiv q \pmod{n}.$$

Otherwise, *q* is called a **quadratic nonresidue** modulo *n*.

OBSERVATIONS ON THE PROTOCOL

Multiple rounds

Each round consists of 3 steps

- Commit; challenge; respond

If challenge can be predicted, then cheating is possible.

- Cannot convince a third party (even if the party is online)
- Essense why it is ZK

If respond to more than one challenge with one commit, then the secret is revealed.

- Essence that this proves knowledge of the secret

ANALYSIS OF THE FIAT-SHAMIR PROTOCOL

Completeness: when proven is given $w^2=x$ and both parties follow the protocol, the verification succeeds

Soundness: if x is not QR, verifier will not be fooled.

- Needs to show that no matter what the prover does, the verifier's verification fails with some prob. (1/2 in this protocol)
- Assumes that x is not QR, V receives y
  - Case 1: y is QR, then when b=1, checking $z^2=yx$ will fail.
  - Case 2: y is QNR, then when b=0, checking $z^2=y$ will fail.
  - Proof will be rejected with probability ½.

FORMALIZING ZK PROPERTY

A protocol is ZK if a simulator exists

- Taking what the verifier knows before the proof, can generate a communication transcript that is indistinguishable from one generated during ZK proofs
  - Intuition: One observes the communication transcript. If what one sees can be generated oneself, one has not learned anything new knowledge in the process.

Three kinds of indistinguishability

- Perfect (information theoretic)
- Statistical
- Computational

FIAT-SHAMIR IS HONEST-VERIFIER ZK

The transcript of one round consists of

- (n, x, y, b, z) satisfying $z^2=yx^b$
- The bit b is generated by honest Verifier V is uniform independent of other values

Construct a simulator for one-round as follows

- Given (x,n)
- Pick at uniform random $b \leftarrow \{0,1\}$,
- If b=0, pick random z and sets $y=z^2 \bmod n$
- If b=1, pick random z, and sets $y=z^2x^{-1} \bmod n$
- Output (n,x,y,b,z)

The transcript generated by the simulator is from the same prob. distribution as the protocol run

FIAT-SHAMIR IS ZK

Given any possible verifier V*, A simulator works as follows:

1. Given (x,n) where x is QR; let T=(x,n)
2. Repeat steps 3 to 7 for
3. Randomly chooses b $\leftarrow$ {0,1},
4. When b=0, choose random z, set $y=z^2 \bmod n$
5. When b=1, choose random z, set $y=z^2x^{-1} \bmod n$
6. Invoke let b'=V*(T,y), if $b' \neq b$, go to step 3
7. Output (n,x,y,b,z); T.append((n,x,y,b,z));

Observe that both z^2 and z^2x^{-1} are a random QR; they have the same prob. distribution

ZERO KNOWLEDGE PROOF OF KNOWLEDGE

A ZKP protocol is a proof of knowledge if it satisfies a stronger soundness property:

- The prover must know the witness of the statement

Soundness property: If a prover A can convince a verifier, then a knowledge exactor exists

- A polynomial algorithm that given A can output the secret

The Fiat-Shamir protocol is also a proof of knowledge:

KNOWLEDGE EXTRACTOR FOR THE QR PROTOCOL

If A can convince V that x is QR with probability significantly over ½, then after A outputs y, then A can pass when challenged with both 0 and 1.

Knowledge extractor

- Given an algorithm A that can convince a verifier,
- After A has sent y, first challenge it with 0, and receives z_1 such that $z_1^2=y$
- Then reset A to the stage after sending y, challenge it with 1 and receives z_2 such that $z_2^2=xy$, then compute $s=z_1^{-1}z_2$, we have $s^2=x$

RUNNING IN PARALLEL

All rounds in Fiat-Shamir can be run in parallel

1. Prover: picks random $r_1,r_2,\ldots,r_t$, sends $y_1=r_1^2$, $y_2=r_2^2$, …, $y_t=r_t^2$
2. Verifier checks the y's are not 0 and sends t random bits $b_1,\ldots b_t$
3. Prover sends $z_1,z_2,\ldots,z_k$,
4. Verifier accept if $z_j^2 \equiv y_j x^{b\_j}$ mod n

This protocol is still a proof of knowledge.

This protocol is still honest verifier ZK (simulator exists for the verifier algorithm V).

It is unknown <u>whether this protocol is ZK (any algorithm V* that can play the role of verifier) or not</u>!

- Consider the V* such that V* chooses $b_1,\ldots b_t$ to be the first t bits of $H(y_1,y_2,\ldots,y_t)$, where H is a cryptographic hash function.
  - The above method for generating an indistinguishable transcript no longer works.

SCHNORR ID PROTOCOL (ZK PROOF OF DISCRETE LOG)

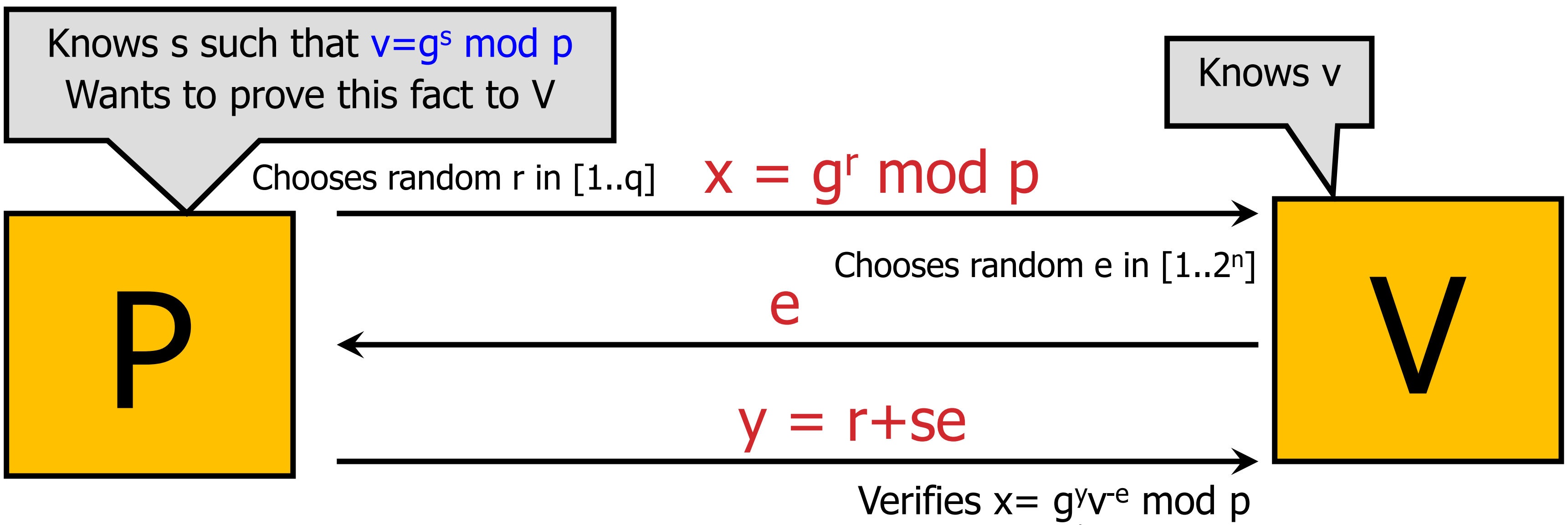
- **System parameter:** **p, q, g**
  - $q|(p-1)$ and g is a generator of Z_p^*
- **Public identity:** **v**
- **Private authenticator:** **s** **$v = g^s \bmod p$**
- **Protocol (proving knowledge of discrete log of v with base g)**

1. A: picks random r in [1..q], sends $x = g^r \bmod p$,
2. B: sends random challenge e in $[1..2^t]$
3. A: sends $y=r+se \bmod q$
4. B: accepts if $x = (g^y v^{-e} \bmod p)$

SCHNORR ID PROTOCOL (ZK PROOF OF DISCRETE LOG)

System parameters

- Prime p and q such that q divides p-1
- g is a generator of an order-q subgroup of Z_p^*

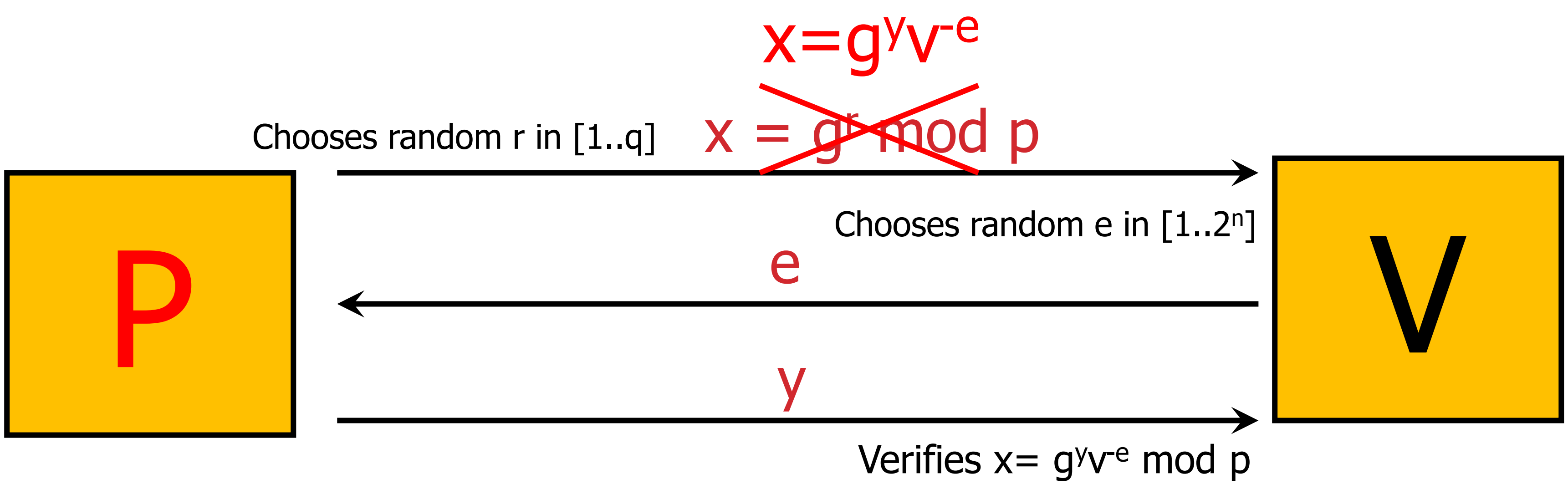


P proves that he knows discrete log of v without revealing its value

CHEATING SENDER

Prover can cheat if he can guess e in advance

- Guess e, set $x=g^y v^{-e}$ for random y in 1st message
- What is the probability of guessing e?

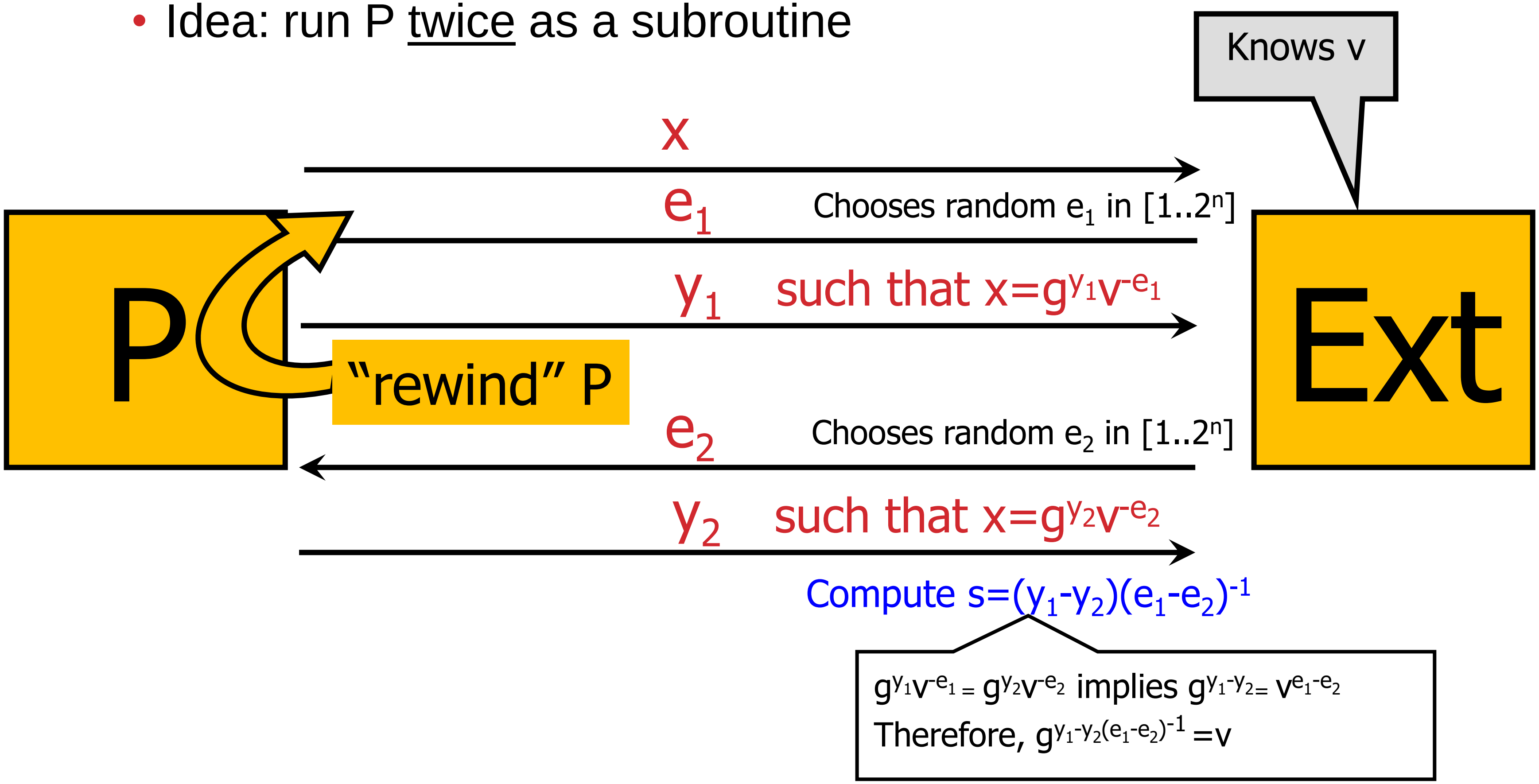


P proves that he "knows" discrete log of v even though he does not know s

SCHNORR'S ID PROTOCOL IS SOUND

Given P who successfully passes the protocol, extract s such that $v=g^s \bmod p$

- Idea: run P twice as a subroutine



SECURITY OF SCHNORR ID PROTOCOL

Completeness: Straightforward

Probability of forgery: $1/2^t$

Soundness (proof of knowledge):

- if A can successfully answer two challenges e_1 and e_2, i.e., A can output y_1 and y_2 such that $x=g^{y1}v^{-e1}=g^{y2}v^{-e2}$ (mod p) then $g^{(y1-y2)}=v^{(e1-e2)}$ and thus the secret

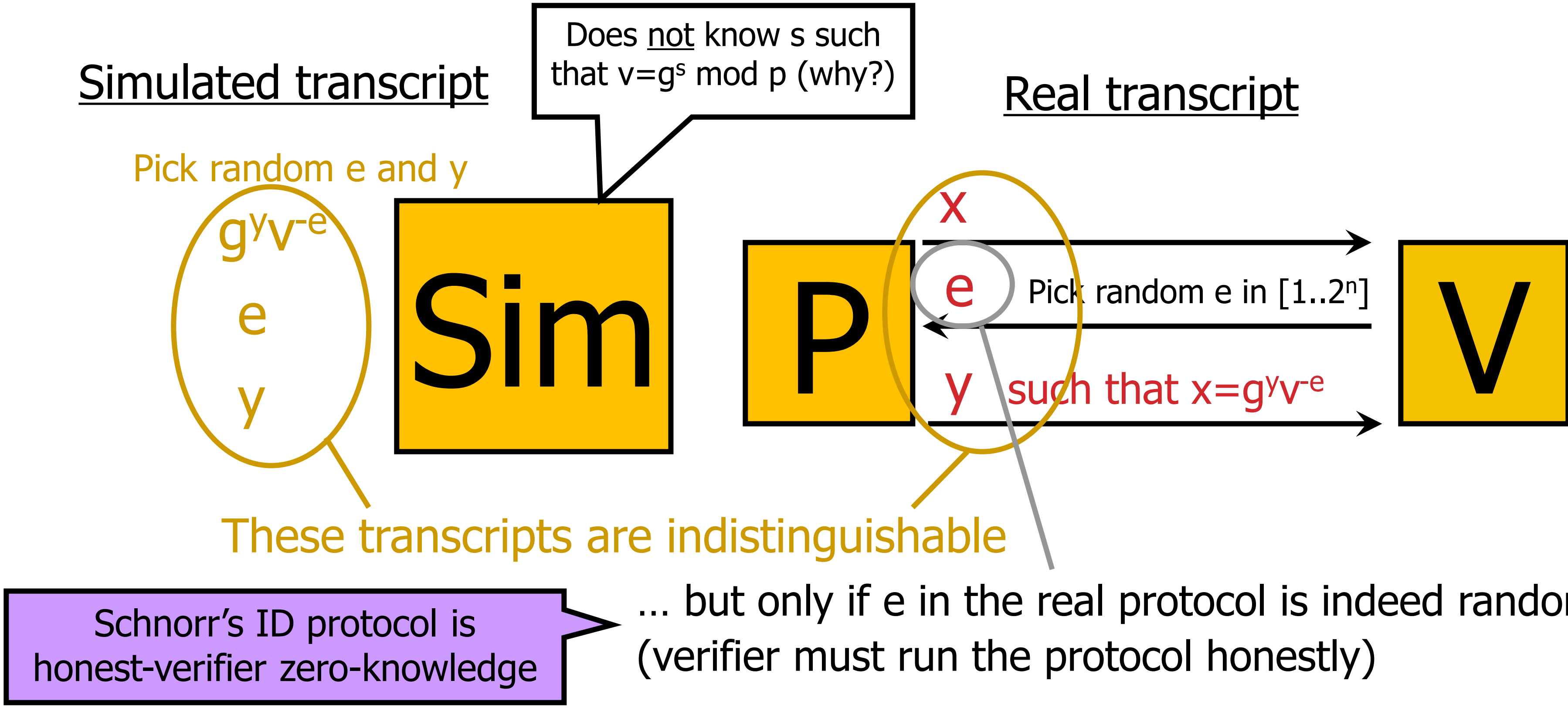
  $s=(y_1-y_2)(e_1-e_2)^{-1}$ mod q

ZK property

- Is honest verifier ZK
- Is ZK when t is small

SCHNORR'S ID PROTOCOL IS HONEST VERIFIER ZK

Simulator produces a transcript which is indistinguishable from the real transcript



… but only if e in the real protocol is indeed random (verifier must run the protocol honestly)

SCHNORR'S ID PROTOCOL IS NOT ZK

ILLINOIS INSTITUTE OF TECHNOLOGY

Schnorr's ID protocol is <u>not</u> zero-knowledge for malicious verifier if challenge e is large

P $\xrightarrow{x}$ V

P $\xleftarrow{e}$ V   Pick some e (may depend on x)

P $\xrightarrow{y}$ V   such that $x=g^y v^{-e}$

Triple (x,e,y) is a solution to the equation $x=g^y v^{-e}$

Verifier may not be able to come up with such a triple on his own.
Therefore, he learned something from the protocol
(protocol is not zero-knowledge!)

OUTLINE

- Zero Knowledge Proof
- Cryptographic Commitment

COMMITMENT SCHEMES

ILLINOIS INSTITUTE OF TECHNOLOGY

An electronic way to temporarily hide a value that cannot be changed

- Stage 1 (Commit)
  - Sender locks a message in a box and sends the locked box to another party called the Receiver
- Stage 2 (Reveal)
  - the Sender proves to the Receiver that the message in the box is a certain message

SECURITY PROPERTIES OF COMMITMENT SCHEMES

ILLINOIS INSTITUTE OF TECHNOLOGY

Hiding

- at the end of Stage 1, no adversarial receiver learns information about the committed value

Binding

- at the end of Stage 1, no adversarial sender can successfully convince reveal two different values in Stage 2

A BROKEN COMMITMENT SCHEME

Using encryption

- Stage 1 (Commit)
  - the Sender generates a key *k* and sends $E_k[M]$ to the Receiver
- Stage 2 (Reveal)
  - the Sender sends *k* to the Receiver, the Receiver can decrypt the message

What is wrong using the above as a commitment scheme?

FORMALIZING SECURITY PROPERTIES OF COMMITMENT SCHEMES

ILLINOIS INSTITUTE OF TECHNOLOGY

Two kinds of adversaries

- those with infinite computation power and those with limited computation power

Unconditional hiding

- the commitment phase does not leak any information about the committed message, in the information theoretical sense (similar to perfect secrecy)

Computational hiding

- an adversary with limited computation power cannot learn anything about the committed message (similar to semantic security)

FORMALIZING SECURITY PROPERTIES OF COMMITMENT SCHEMES

ILLINOIS INSTITUTE OF TECHNOLOGY

Unconditional binding

- after the commitment phase, an infinite powerful adversary sender cannot reveal two different values

Computational binding

- after the commitment phase, an adversary with limited computation power cannot reveal two different values

No commitment scheme can be <u>both unconditional hiding and unconditional binding</u>

ANOTHER (ALSO BROKEN) COMMITMENT SCHEME

Using a one-way function *H*

- Stage 1 (Commit)
  - the Sender sends c=$H(M)$ to the Receiver
- Stage 2 (Reveal)
  - the Sender sends M to the Receiver, the Receiver verifies that c=$H(M)$

What is wrong using this as a commitment scheme?

A workable scheme (though cannot prove security)

- Commit: choose r1, r2, sends (r1, H(r1||M||r2))
- Reveal (open): sends M, r2.
- Disadvantage: Cannot do much interesting things with the commitment scheme.

PEDERSEN COMMITMENT SCHEME

Setup: receiver chooses…

- Large primes p and q such that q divides p-1
- Generator g of the order-q subgroup of Z_p^*
- Random secret a from Z_q
- $h=g^a \bmod p$
  - Values p,q,g,h are public, a is secret

Commit: to <u>commit to some $x \in Z_q$</u>, sender chooses random $r \in Z_q$ and sends $c=g^x h^r \bmod p$ to receiver

- This is simply $g^x(g^a)^r=g^{x+ar} \bmod p$

Reveal: to open the commitment, sender reveals x and r, receiver verifies that $c=g^x h^r \bmod p$

PEDERSEN COMMITMENT SCHEME (CONT.)

ILLINOIS INSTITUTE OF TECHNOLOGY

Unconditionally hiding

- Given a commitment c, every value x is equally likely to be the value committed in c.
- For example, given x, r, and any x', there exists r' such that $g^x h^r = g^{x'} h^{r'}$, in fact $r' = (x-x')a^{-1} + r \bmod q$ (but must know a to compute r')

Computationally binding

- Suppose the sender open another value x' ≠ x.
- That is, the sender find x' and r' such that $c = g^{x'} h^{r'} \bmod p$. Now the sender knows x, r, x', r' s.t., $g^x h^r = g^{x'} h^{r'} \pmod p$, the sender can compute $\log_g(h) = (x'-x)\cdot(r-r')^{-1} \bmod q$.
- Assume DL is hard, the sender cannot open the commitment with another value.

PEDERSEN COMMITMENT – ZK PROVE KNOW HOW TO OPEN (WITHOUT ACTUALLY OPENING)

Public commitment $c = g^x h^r \pmod p$

Private knowledge x, r

Protocol:

1. P: picks random y, s in [1..q], sends

   $d = g^y h^s \bmod p$

2. V: sends random challenge e in [1..q]
3. P: sends $u=y+ex$, $v=s+er \pmod q$
4. V: accepts if $g^u h^v = dc^e \pmod p$

Security property – similar to Schnorr protocol

ACKNOWLEDGMENTS

Note: Some of the slides in this lecture are based on material created by

- Dr. Ninghui Li at Purdue University
- Dr. Vitaly Shmatikov at Cornell