CSC3631 Cryptography - Zero Knowledge

Thomas Gross

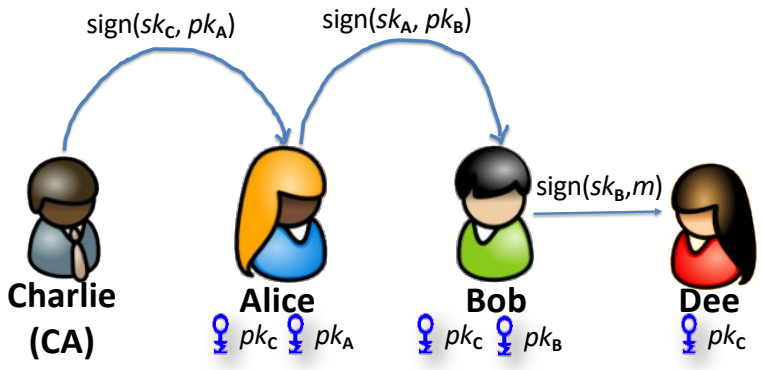
1

Certificate Verification Procedure

1. Acquire the authentic public key pk_C of the CA
2. Obtain an identifying string id_A which uniquely identifies party **A**
3. Acquire over an unsecure channel the public-key certificate pk_A of party **A**, agreeing with the identifying string id_A.
4. Verify:
   a) Current date and time against the validity period of pk_A
   b) Current validity of CA's public key pk_C
   c) Signature on **A**'s certificate using the CA's pk_C
   d) Certificate on pk_A not revoked
5. If all checks succeed, accept pk_A in the certificate as authentic public key.

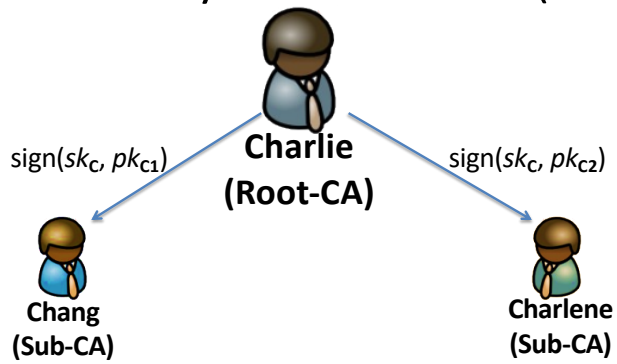
4

How to create a certificate chain?



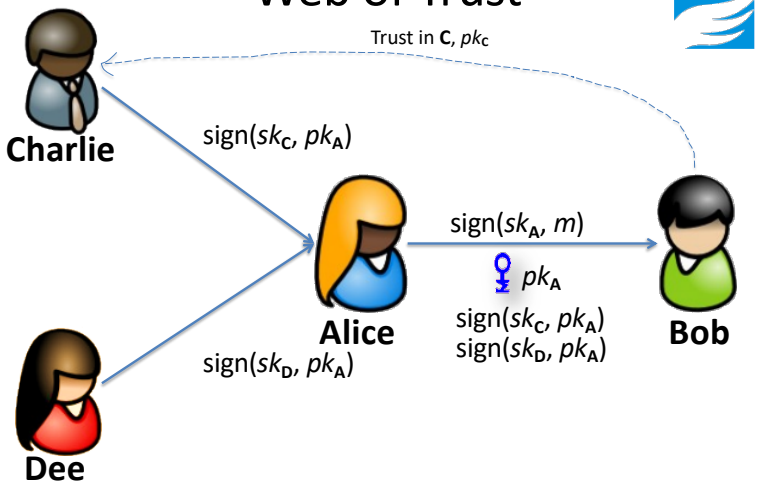
5

Public Key Infrastructure (PKI)



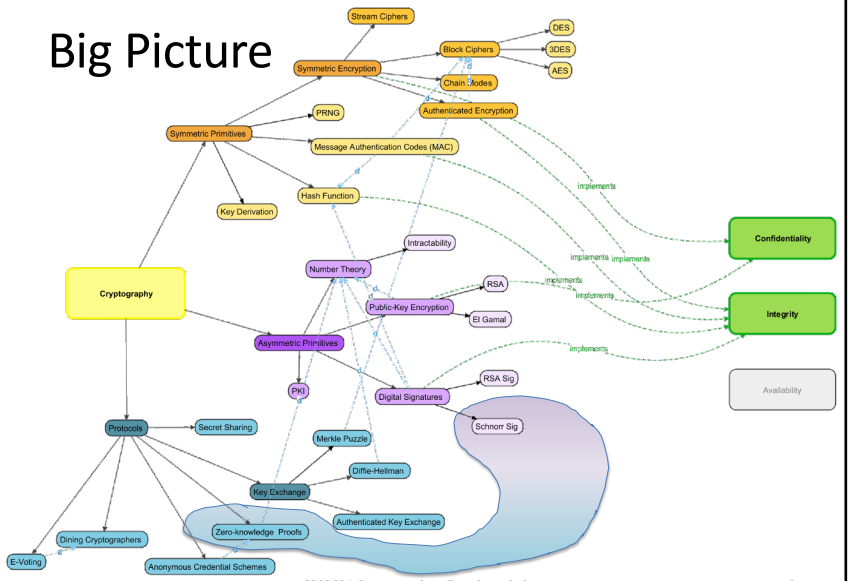
6

Web of Trust



[GPG logo from gnupg.org, as example only. No trademark implications intended.]

Big Picture



Roadmap

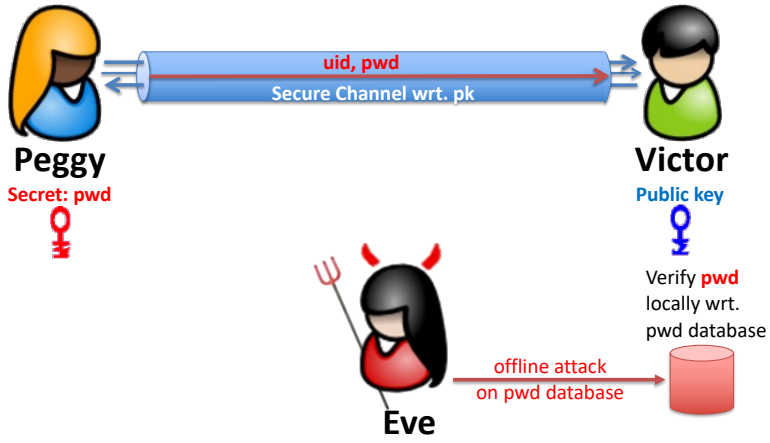
- **The Notion of Zero-Knowledge**
  - A Tale of Ali Baba
  - Interactive Proof Systems
  - Zero-Knowledge Proof of Knowledge
- The Schnorr Identification Protocol

Goal for today:

- **What's the intuition behind zero-knowledge?**
- **How does a simple zero-knowledge protocol work?**

Example: Password-Authentication I

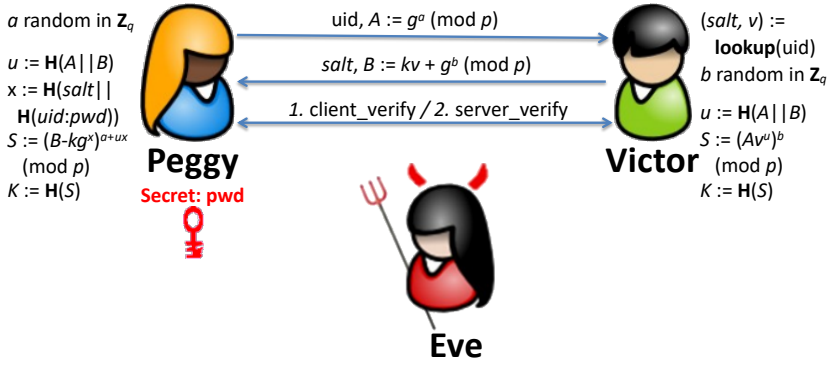
How to authenticate Peggy?



Example: Password-Authentication II

PAKE: Secure Remote Password (SRP) Protocol

Setup: $(\mathbf{Z}_p)^*$, where p prime, G with generator g, prime-order q, $k := \mathbf{H}(g||p)$
random salt **Secret:** $x := \mathbf{H}(salt||\mathbf{H}(uid{:}pwd))$ Public Verifier: $v := g^x \pmod p$



[Wu. The Secure Remote Password protocol (SRP 6a, SRP), 1998, RFC 2945, 5054]

Properties & Problems of SRP

- Widely deployed, especially in OpenSSL
- No security proof exists.
- **There exist precomputation attacks:**
  Standardized SRP does not protect from server leaks, adversary can precompute values wrt. user salt and password dictionary.
- **Timing attack:**
  OpenSSL implementations of SRP attacked with offline dictionary attacks, based on non-constant time execution of a modular exponentiation.

[Braga, Fouque, Sabt. PARASITE: PAssword Recovery Attack against Srp Implementations in ThE wild. 2021]

Problem Statement

How can we prove knowledge of a secret to a verifier without disclosing any information about the secret?

Roadmap

- **The Notion of Zero-Knowledge**
  - **A Tale of Ali Baba**
  - Interactive Proof Systems
  - Zero-Knowledge Proof of Knowledge
- The Schnorr Identification Protocol

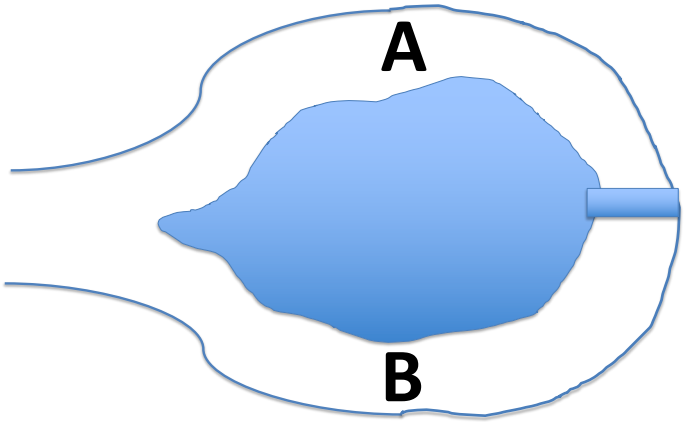
Goal for today:
- **What's the intuition behind zero-knowledge?**
- **How does a simple zero-knowledge protocol work?**

"iftah ya simsim"

[Image from Wikipedia]

16

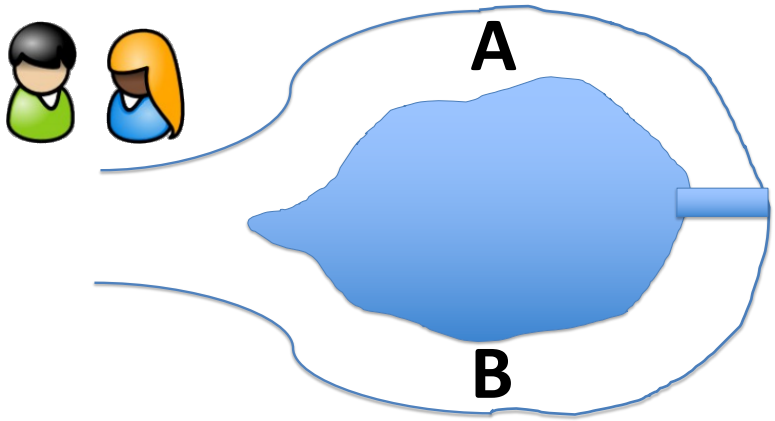
Ali Baba's Cave



[Metaphor and graphic adapted from Quisquater et al., CRYPTO'98]

17

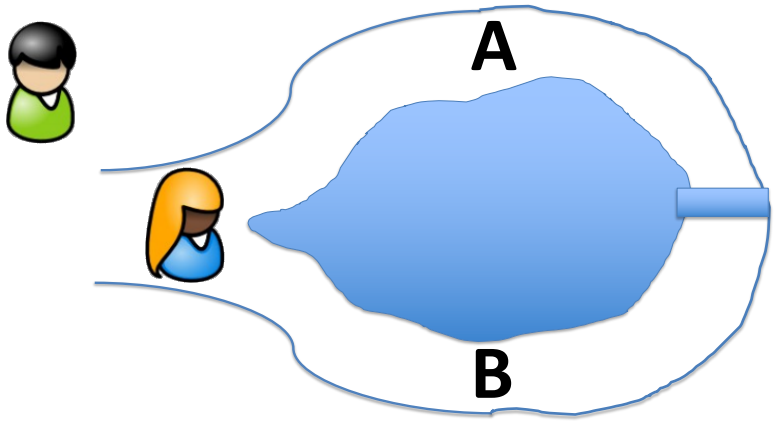
Ali Baba's Cave



[Metaphor and graphic adapted from Quisquater et al., CRYPTO'98]

18

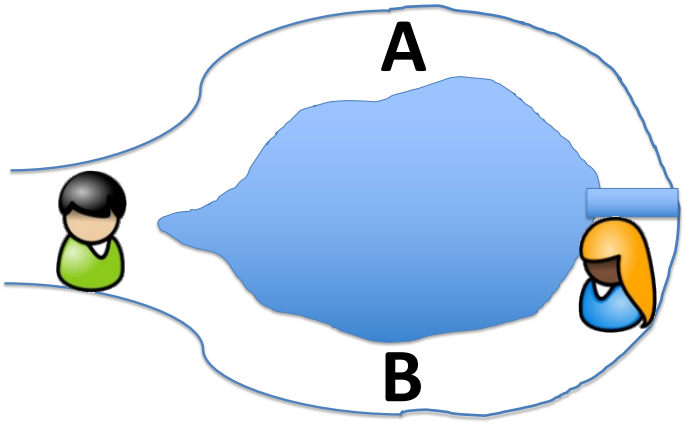
Ali Baba's Cave



[Metaphor and graphic adapted from Quisquater et al., CRYPTO'98]

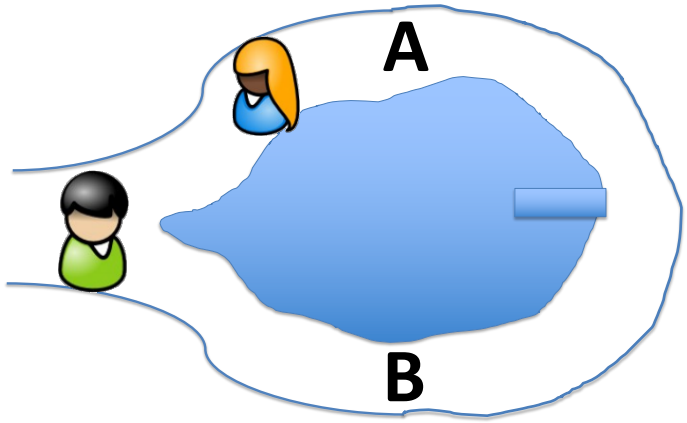
19

Ali Baba's Cave



[Metaphor and graphic adapted from Quisquater et al., CRYPTO'98]

Ali Baba's Cave



[Metaphor and graphic adapted from Quisquater et al., CRYPTO'98]

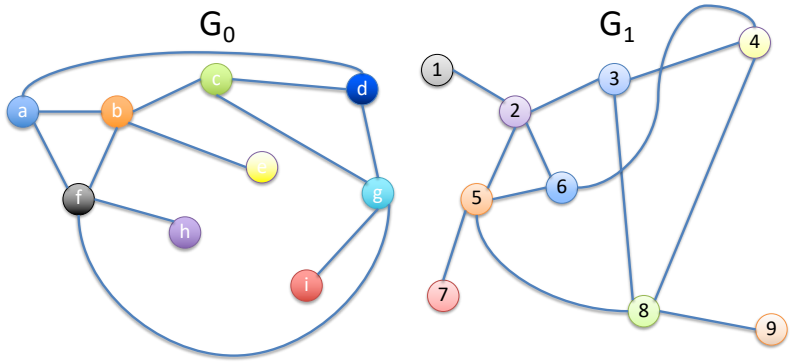
Roadmap

- **The Notion of Zero-Knowledge**
  - A Tale of Ali Baba
  - **Interactive Proof Systems**
  - Zero-Knowledge Proof of Knowledge
- The Schnorr Identification Protocol

Goal for today:

- **What's the intuition behind zero-knowledge?**
- **How does a simple zero-knowledge protocol work?**

Claim: Know Similarity Transformation "Isomorph"

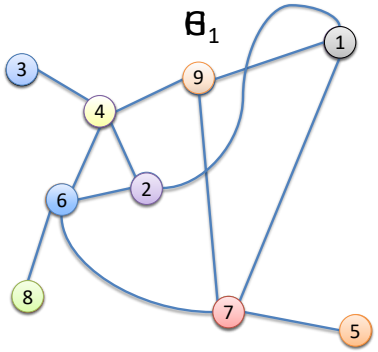


Know ϕ: e = 1 a = 6 b = 2 d = 4 …

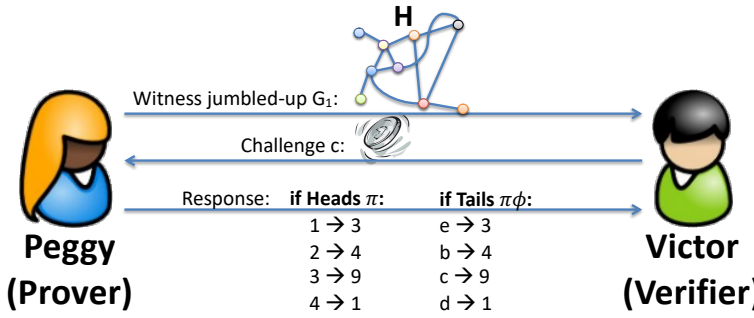
Jumble Up a Graph "Permutation" π

π randomly shuffles graph G_1

1 → 3
2 → 4
3 → 9
4 → 1
5 → 6
6 → 2
7 → 8
8 → 7
9 → 5



Interactive Proof

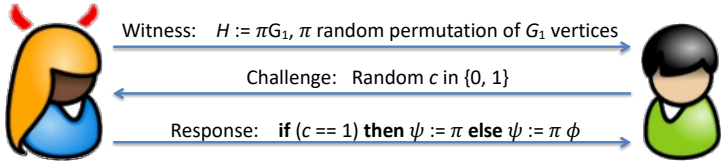


reject if transformation applied to H doesn't match G_c

Interactive Proof Systems

How can the prover convince a verifier interactively?

Example: Peggy knows a graph isomorphism ϕ between G_0 and G_1

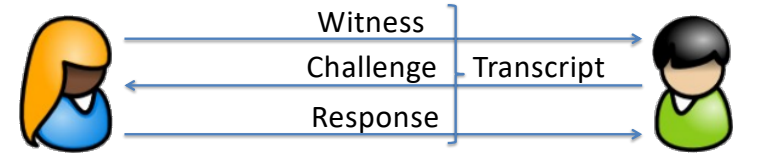


dishonest

accept or **reject**
reject if $H \mathrel{!}= \psi G_c$

Interactive Proof Systems

Focusing on 3-round "Sigma" protocols



accept or **reject**

Proof of Knowledge

How can Peggy prove that she knows a secret?

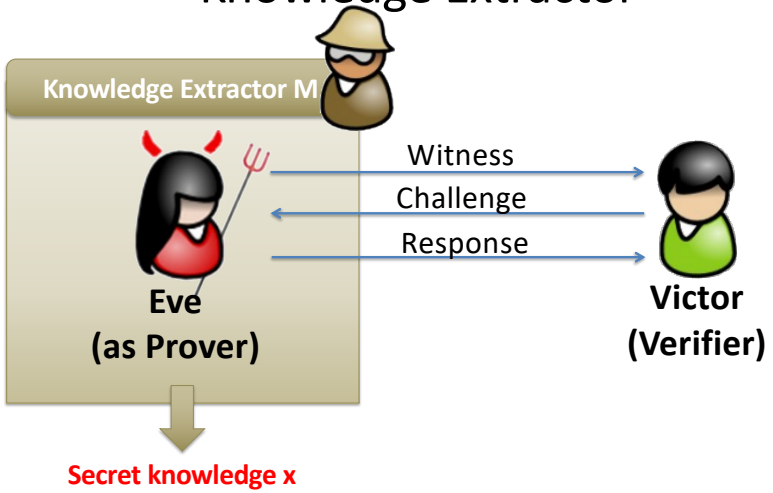
Completeness:

If the prover knows the secret, then the proof succeeds with overwhelming probability.

Soundness:

There exists a polynomial-time **knowledge extractor** M, such that if an adversary can execute the protocol successfully with non-negligible probability, then M can be used to extract from Eve the knowledge to run the protocol successfully.

Knowledge Extractor



Roadmap

- **The Notion of Zero-Knowledge**
  - A Tale of Ali Baba
  - Interactive Proof Systems
  - **Zero-Knowledge Proof of Knowledge**
- The Schnorr Identification Protocol

Goal for today:

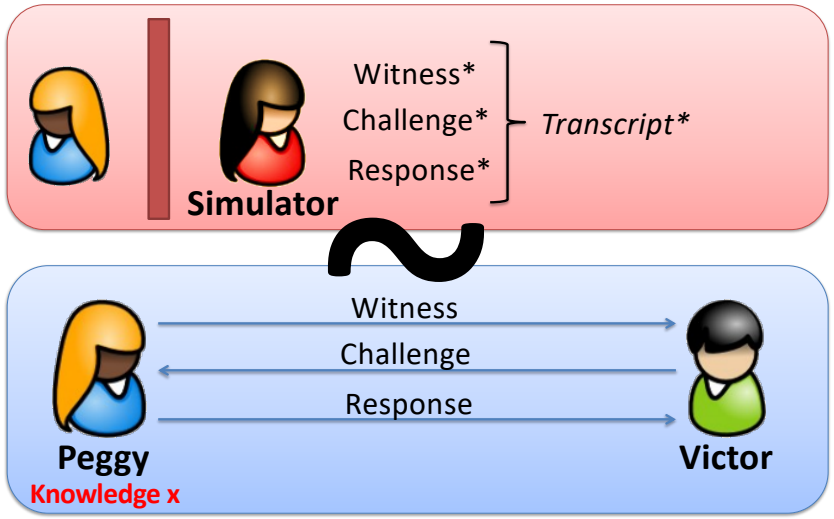
- **What's the intuition behind zero-knowledge?**
- **How does a simple zero-knowledge protocol work?**

Zero-Knowledge Property

Zero-Knowledge:

There exists a polynomial-time **simulator** which can produce upon input of the assertions without access to the real prover, transcripts **indistinguishable** from those resulting from a protocol with a real prover.

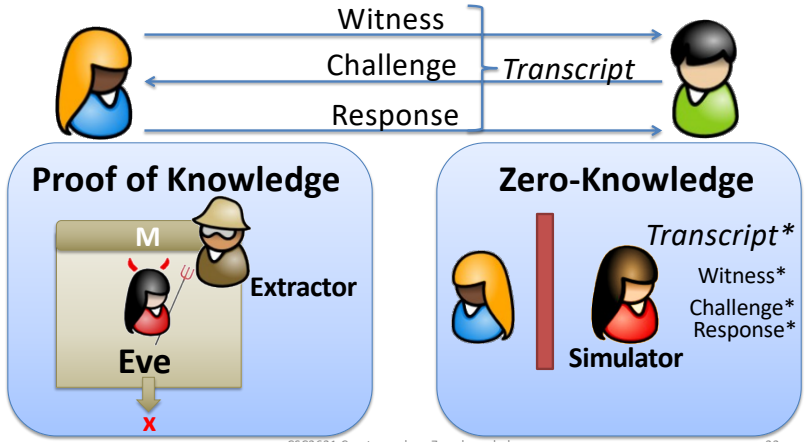
Simulator



32

Summary

Interactive Proofs (3-round):



33

Roadmap

- The Notion of Zero-Knowledge
  - A Tale of Ali Baba
  - Interactive Proof Systems
  - Zero-Knowledge Proof of Knowledge
- **The Schnorr Identification Protocol**

Goal for today:

- **What's the intuition behind zero-knowledge?**
- **How does a simple zero-knowledge protocol work?**

34

Schnorr Key Generation

How to create a strong setting for Schnorr?

GenSchnorr(1^n)

Input: key length n

Create a cyclic group G, sub-group of $(\mathbf{Z}_p)^*$ with generator g with prime-order q.

Choose random x in $\mathbf{Z}_q$

Compute $y = g^x$ (**mod** q)

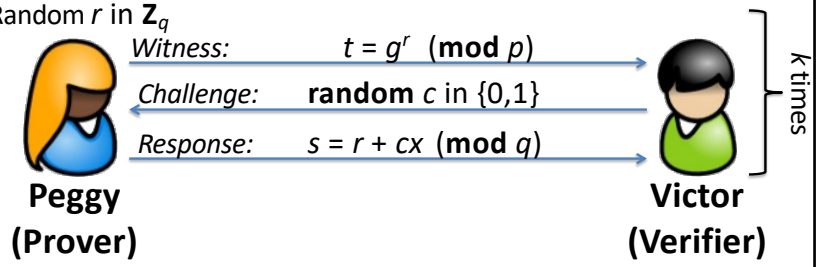
Output: $pk=(G, q, g, y)$, $sk=(G, q, g, x)$

35

Multi-Round Schnorr Protocol

$sk=(G, q, g, x)$ $pk=(G, q, g, y)$

Random r in $\mathbf{Z}_q$

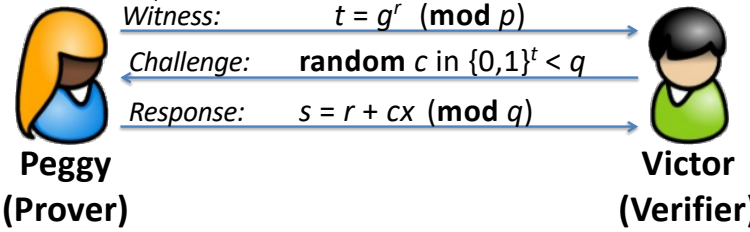


accept only if $g^s \stackrel{?}{=} ty^c \pmod p$

The Schnorr Identification Protocol

$sk=(G, q, g, x)$ $pk=(G, q, g, y)$

Random r in $\mathbf{Z}_q$



accept only if $g^s \stackrel{?}{=} ty^c \pmod p$

Example
Schnorr Identification Protocol

- **Setup:** prime modulus $p = 48731$
  $q = 443$, $g = 11444$
- Private key $x = 357$, public key $y = 45776$
- Witness: $r = 274$, $t = g^r = 37123 \pmod{48731}$
- Challenge: $c = 129$
- Response: $s = r + cx = 255 \pmod{443}$
- Verification: $11444^{255} = 37123\ 45776^{129} \pmod{48731}$

[Menezes. Chapter 10.4.4 Schnorr Identification Protocol. Note: example is changed slightly to match description]

Is this Zero-Knowledge?

Let's build a **simulator**.
(Only considering **honest** prover and verifiers)

Choose	c, s in $\mathbf{Z}_q$	
Compute	$t = g^s / y^c$	**(mod** p**)**
Output	(t, c, s)	*transcript*

Schnorr is **Honest-Verifier Zero-Knowledge.**

Example Simulation

- Same setup as previous example
- Choose *c*, *s* first:
  - *c* := 129
  - *s* := 255
- Compute t := g^s / y^c (**mod** *p*)
  - *t* = 11444^{255} / 45776^{129} (**mod** 48731)
  - *t* = 11444^{255} $(12806)^{-1}$ (**mod** 48731)
  - *t* = 11444^{255} 10575 (**mod** 48731) = 37123

Schnorr is also a Signature Scheme

Transformation of an Interactive Proof with Fiat-Shamir

Sign:

Choose random *r*; $t = g^r$ (**mod** *p*)
Compute $c = \mathbf{H}(m||t)$
Compute $s = r - cx$ (**mod** *q*)
Signature: $\sigma = (s, c)$

Verify:

Compute $t_V = g^s y^c$ (**mod** *p*)
Check $c \stackrel{?}{=} \mathbf{H}(m||t_V)$

Zero-Knowledge Proof Notation

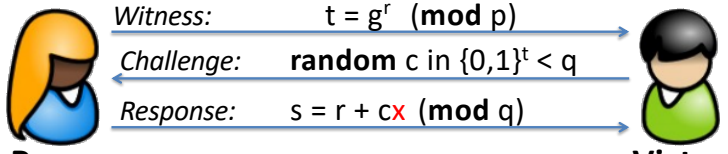
We can prove knowledge of **linear equations in the exponent**.

To **prove knowledge** of a secret x and a relation to a public y, we write:

PK {(*x*):
$y = g^x$ (mod *p*)
}

Translates to a Schnorr Proof…

sk=(G, q, g, x) *pk*=(G, q, g, y)



PK {(*x*):
$y = g^x$ (mod *p*)
}

accept only if $g^s \stackrel{?}{=} t y^c$

… and a Signature Scheme

(Fiat-Shamir)

sk=(G, q, g, x) pk=(G, q, g, y)

Witness: $t = g^r \pmod p$

Challenge: "random" $c = H(m||t)$

Response: $s = r - cx \pmod q$

Peggy (Signer)

SPK {(x): $y = g^x \pmod p$ }

Victor (Verifier)

accept only if

$t \stackrel{?}{=} g^s y^c$

$c \stackrel{?}{=} \mathbf{H}(m||t)$

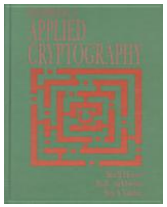
Summary

Schnorr protocol as simple zero-knowledge protocol.

Only zero-knowledge in presence of an **honest verifier**.

Can be transformed to a signature with **Fiat-Shamir heuristic**.

Literature



Menezes et al.

Handbook of applied cryptography.

1997

http://cacr.uwaterloo.ca/hac/

Section 10.4.1 Overview of zero-knowledge concepts

Section 10.4.4 The Schnorr identification protocol