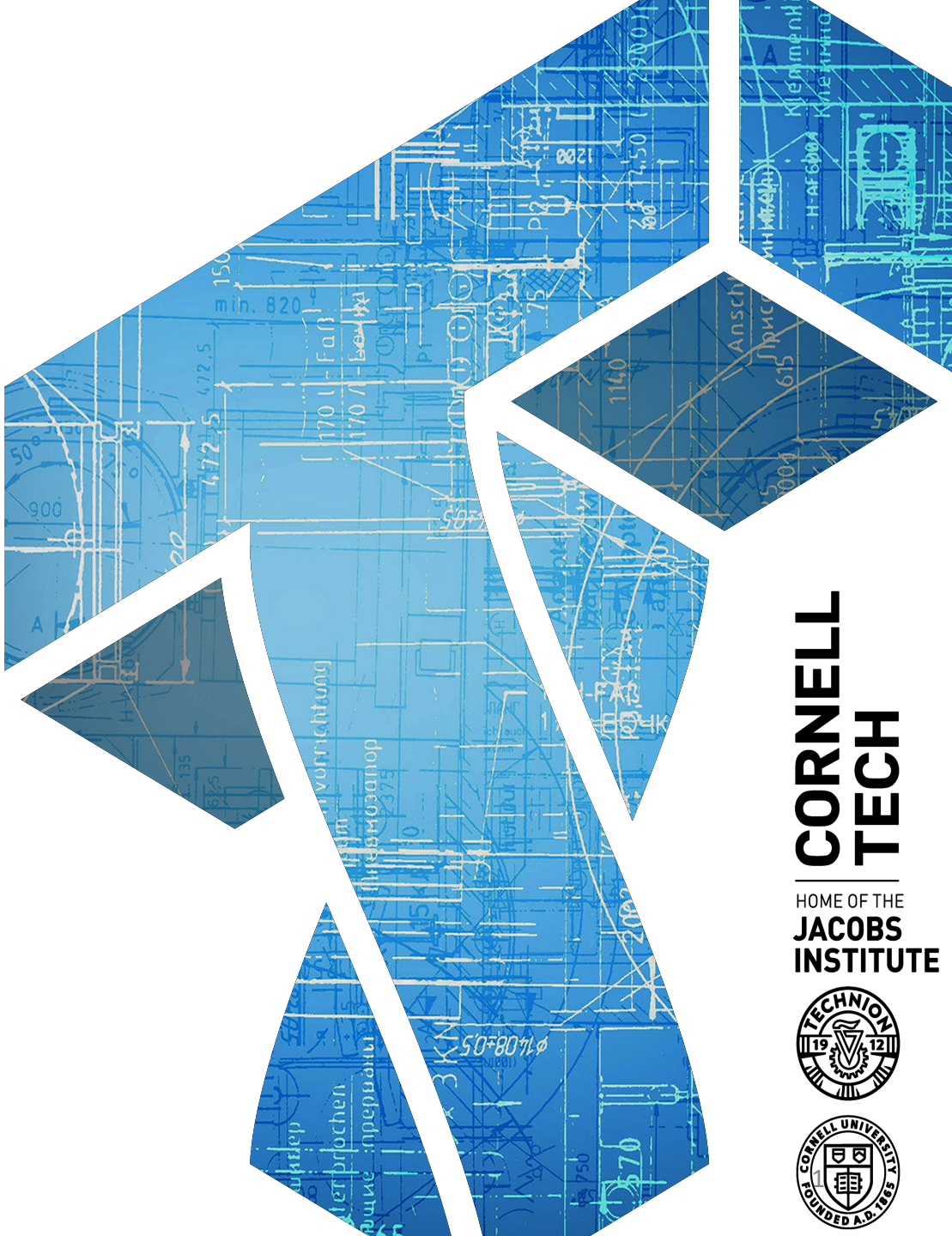
CS 5830
Cryptography

Instructor: Tom Ristenpart
TAs: Yan Ji, Sanketh Menda

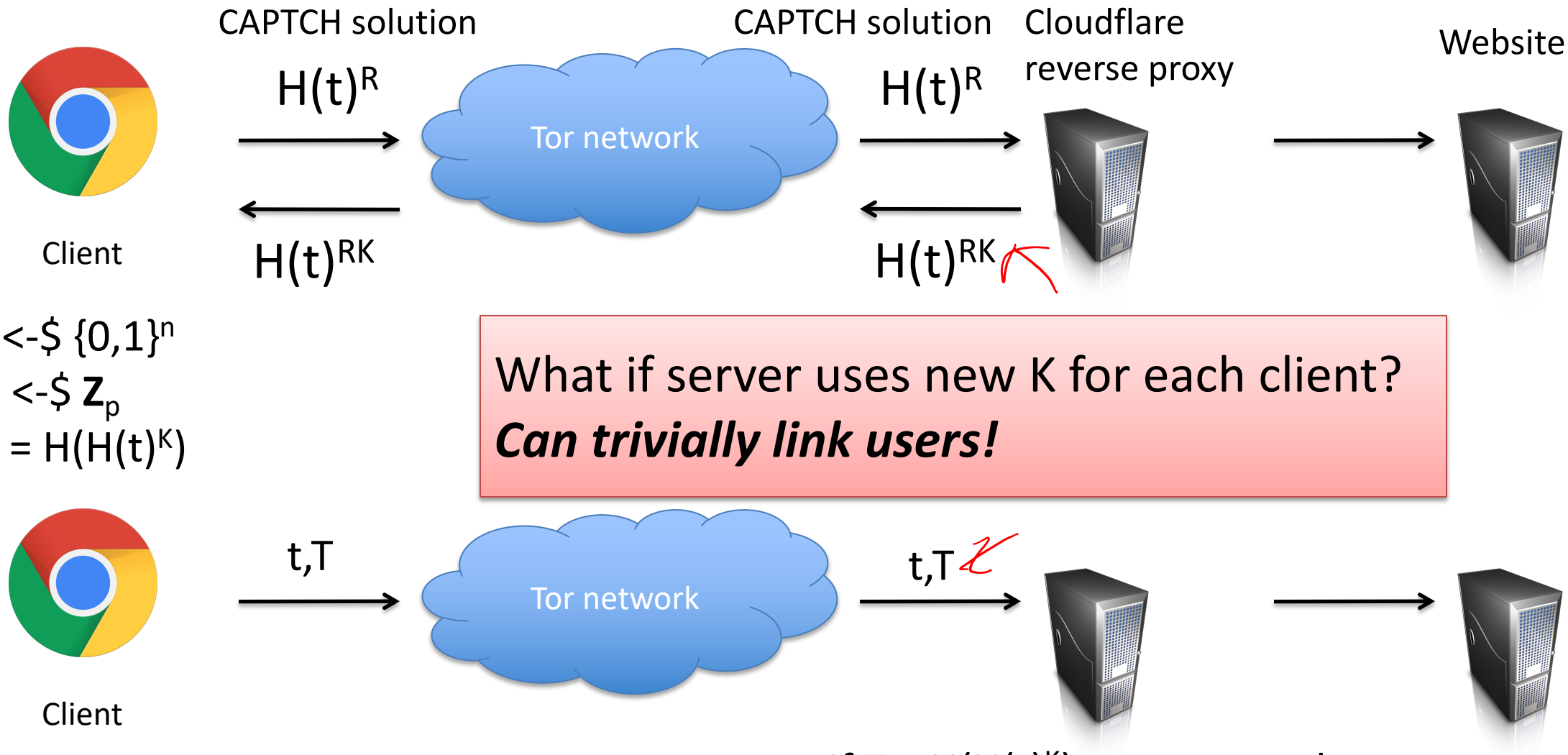


CORNELL TECH
HOME OF THE JACOBS INSTITUTE

Privacy Pass

- OPRF-based solution written up in Privacy Pass paper
- Subsequently standardized and deployed
- One concern is ***hoarding attacks***
  - Attacker harvests a bunch of tokens
  - Attacker spends all tokens at once to perform DDoS attack
  - Unlinkability prevents detection
  - Solution: partially oblivious PRFs
    - F_K(time,x) where time is public input, x is private input
    - Problem: no fast PO-PRFs were known
- Another concern is ***tagging attacks by proxy***

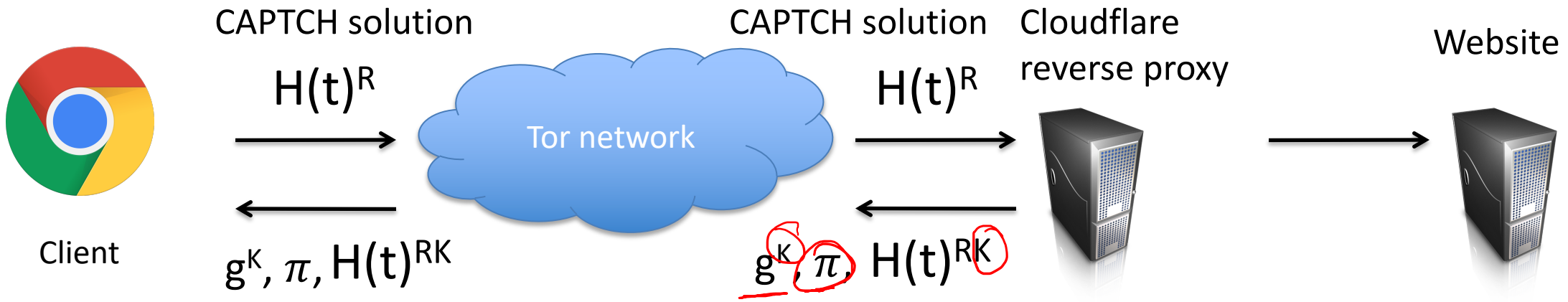
Privacy Pass



Resisting active tagging attacks

- Privacy Pass: proxy needs to prove that they sign each request with the same secret key K
  - Can publish a public key $pk = g^K$
  - Prove that Y^K and pk have same discrete log, without revealing K
- Zero-knowledge proofs (ZKPs) can handle this

Privacy Pass



$t \leftarrow\$ \{0,1\}^n$
$R \leftarrow\$ \mathbf{Z}_p$
$T = H(H(t)^K)$

Verify π
- If fails, then abort
- Otherwise can use tokens

This is called a ***verifiable OPRF***. In words, it allows ensuring consistency (relative to public key g^K) of server behavior

How do prove that $dlog_g(g^K) = dlog_Y(Y^K)$ where $Y = H(t)^R$?

Naïve approach: Send K!

Why is that problematic?

Zero-knowledge proofs

- Prove knowledge of secret x without revealing any information about x
- Introduced by Goldwasser, Micali, Rackoff
  - "The Knowledge Complexity of Interactive Proof-Systems", 1989
- Relations: $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{Y}$

Witnesses

secret

Statements

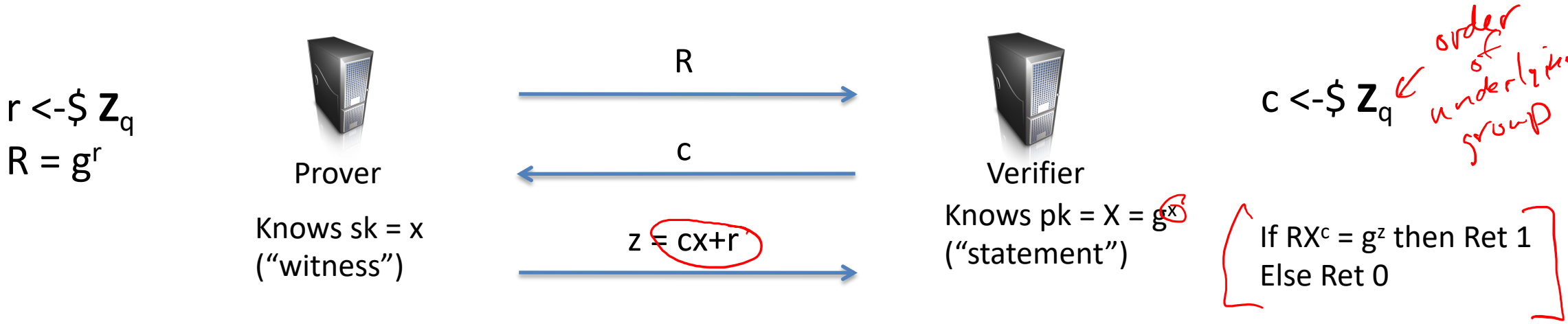
public key

- Sigma protocols classic approach for ZKP

Zero-knowledge properties

- ***Completeness***: can prove a true statement
- ***Soundness***: can't prove a false statement
  - *ZK proofs*: no unbounded adversary can prove false statement
  - *ZK arguments*: no computationally bounded adversary can prove false statement
  - *Knowledge soundness*: to prove statement, prover must know a witness
- ***Zero-knowledge***: proof reveals nothing (except validity)
  - *Perfect ZK*: can simulate proofs with same distribution as real
  - *Statistical ZK*: simulated proofs statistically indistinguishable from real
  - *Computational ZK*: simulated proofs computationally indistinguishable from real
  - *Concurrent ZK*: simulation despite interleaved proof execution
- **Interactive vs. non-interactive**
  - NIZK (non-interactive ZK proof) allows prover to generate a single value that convinces verifier.
  - Interactive requires multiple back-and-forths between prover and verifier

Schnorr Sigma protocol



Prover wants to prove that they have sk = x associated to public key $X = g^x$

Sigma protocols are class of 3-round protocols:

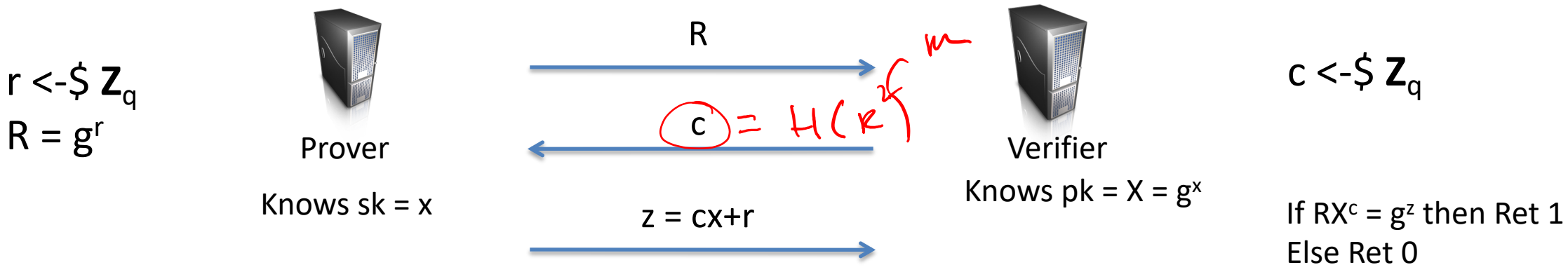
- Prover sends ***commitment***
- Verifier replies with ***challenge***
- Prover sends ***response***

$RX^c = g^r \cdot (g^x)^c = g^{r+xc} = g^z$

Schnorr just one example, many others:

- Okamoto's, Chaum-Pederson, Guillou-Quisquater (RSA), …

Fiat-Shamir transform



Signature is commitment and response. Challenge generated non-interactively with hash function
Works on other Sigma protocols as well

Sign(x, M)

$r \leftarrow\$ Z_q$

$R = g^r$; $c = H(M \,||\, R)$; $z = r + cx \mod q$

Return (R,z)

Ver(X, M, (R,z))

$c = H(M \,||\, R)$

If $g^z = RX^c$ then Return 1

Return 0

(Variant of) Schnorr signatures

Let G be group of prime order q. Let g be a generator

sk = x chosen randomly from $\mathbf{Z}_q$ pk = X = g^x

Sign(x, M)

r <-$ $\mathbf{Z}_q$

R = g^r ; c = H(M || R) ; z = r + cx mod q

Return (R,z)

Ver(X, M, (R,z))

c = H(M || R)

If $g^z = RX^c$ then Return 1

Return 0

Correctness? $g^z = g^{r + cx} = g^r g^{xc} = RX^c$

Incorrect version of Chaum-Pederson Sigma protocol (DL equality!)

$r \leftarrow_\$ \mathbb{Z}_p$

$R_1 \leftarrow g^r$

$R_2 \leftarrow Y^r$

Prover

Knows x s.t. $X = g^x$

R_1, R_2

c

z

Verifier

Knows X, Y, Z, g

c <-$ $\mathbb{Z}_q$

$z \leftarrow r + xc$

Check both that:

$g^z = R_1 \cdot X^c$

$g^z = R_2 \cdot Y^c$

Verifier wants to check that $\text{dlog}_g(X) = \text{dlog}_Y(Z)$

In our verifiable OPRF application: $X = g^K$, $Y = H(t)^R$, $Z = H(t)^{RK}$

Can apply Fiat-Shamir here as well: c = H(R,X,Y, Z)

Chaum-Pederson Sigma protocol (DL equality!)

$r \leftarrow_{\$} \mathbb{Z}_p$

$R_1 \leftarrow g^r$

$R_2 \leftarrow Y^r$

Prover

Knows x s.t. $X = g^x$

R_1, R_2

c

z

Verifier

Knows X, Y, Z, g

c <-$ $\mathbf{Z}_q$

$z \leftarrow r + xc$

Check both that:

$$g^z = R_1 \cdot X^c$$

$$Y^z = R_2 \cdot Z^c$$

Verifier wants to check that $\text{dlog}_g(X) = \text{dlog}_Y(Z)$

$$Y^z = Y^{r+xc} = Y^r \cdot Y^{xc} = R_2 \cdot Z^c$$

Chaum-Pederson Sigma protocol (DL equality!)

$r \leftarrow_{\$} \mathbb{Z}_p$

$R_1 \leftarrow g^r$

$R_2 \leftarrow Y^r$

Prover

Knows x s.t. $X = g^x$

R_1, R_2

c

z

Verifier

Knows X, Y, Z, g

c <-$ $\mathbf{Z}_q$

$z \leftarrow r + xc$

Check both that:

$$g^z = R_1 \cdot X^c$$

$$Y^z = R_2 \cdot Z^c$$

Verifier wants to check that $\text{dlog}_g(X) = \text{dlog}_Y(Z)$

In our verifiable OPRF application: $X = gK$, $Y = H(t)^R$, $Z = H(t)^{RK}$

Can apply Fiat-Shamir here as well: c = H(R,X,Y, Z)

Security analysis: game plan

- Formalize ID protocols and their security under ***passive attacks***
- Show Schnorr ID protocol secure
  - Uses fact that Schnorr protocol is ***honest-verifier zero-knowledge***
  - Requires so-called ***rewinding lemma***
    - Run an adversary twice on related randomness, likely to succeed twice
- Fiat-Shamir analysis: show that Schnorr signature UF-CMA is implied by Schnorr ID passive attack security
  - Similar analysis shows soundness of NIZK

ID protocols

$r \leftarrow\$ \mathbb{Z}_q$

$R = g^r$

$z = cx + r$

Prover

Knows sk = x

R

c

z = cx+r

Verifier

Knows pk = $X = g^x$

$c \leftarrow\$ \mathbb{Z}_q$

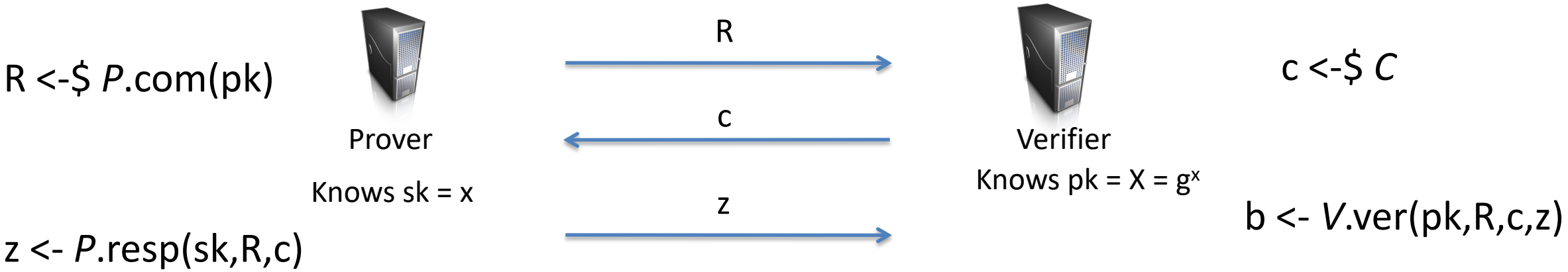
If $RX^c = g^z$ then

    Ret 1

Else

    Ret 0

ID protocols



ID protocol is a tuple of algorithms (IDKg, P.Com,P.Resp,V.ver) and challenge space C
IDKg generates key pair (public key, secret key)

ID security under passive attack

Prover given pk
And transcripts
of q_{id} honest
executions

R*

c*

z*

Verifier

b <- *V*.ver(pk,R,c,z)

For Schnorr:

$R^* X^{c^*} = g^{z^*}$

$\underline{\text{IDPASS}^{\mathcal{A}}_{\text{ID},q_{id}}}$

$(pk, sk) \leftarrow\$ \text{IDkg}$

queried $\leftarrow$ false

For $i = 1$ to q_{id} do

$R_i \leftarrow\$ P.com(pk)$

$c_i \leftarrow\$ \mathcal{C}$

$z_i \leftarrow\$ P.resp(sk, R_i, c_i)$

$z^* \leftarrow\$ \mathcal{A}^{\text{Ver}}(pk, (R_1, c_1, z_1), \ldots, (R_{q_{id}}, c_{q_{id}}, z_{q_{id}}))$

Ret $V.ver(pk, R^*, c^*, z^*)$

$\underline{\text{Ver}(R^*)}$

If queried = true then Ret $\perp$

queried $\leftarrow$ true

$c^* \leftarrow\$ \mathcal{C}$

Ret c^*

Intuition for security:

1. Can simulate R_i , c_i , z_i because Schnorr is zero-knowledge

2. Build DL-adversary *B* that runs *A* twice to get two transcripts that allow extracting sk = x

$$\mathbf{Adv}^{\text{idpass}}_{\text{ID},q_{id}}(\mathcal{A}) = \Pr\left[\text{IDPASS}^{\mathcal{A}}_{\text{ID},q_{id}} \Rightarrow \text{true}\right]$$

Zero-knowledge security

A ***simulator*** Sim is an algorithm that takes an input any (valid) pk and outputs transcript R*, c*, z*

ID-based protocol ID = (P.com, P.resp, V.ver, C) is ***honest-verifier zero-knowledge*** (HVZK) if we can give Sim that produces transcripts distributed identically to ones generated by ID

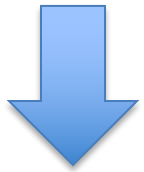
Schnorr is honest-verifier zero-knowledge

How can we create believable transcripts without secret key for pk = X = g^x

Real transcript:		Faked transcript:
g^r	⟷	R = $g^z X^{-c}$
c		c
z = cx + r		z

Sim(pk):
c <-$ C
z <-$ $\mathbf{Z}_p$
R <- g^z pk^{-c}
Return (R,c,z)

Sim works without secret key x



Transcript leaks nothing about x

$\underline{\mathrm{IDPASS}^{\mathcal{A}}_{\mathsf{ID},q_{id}}}$
$(pk, sk) \leftarrow\$ \mathsf{IDkg}$
queried $\leftarrow$ false
For $i = 1$ to q_{id} do
  $R_i \leftarrow\$ P.com(pk)$
  $c_i \leftarrow\$ \mathcal{C}$
  $z_i \leftarrow\$ P.resp(sk, R_i, c_i)$
$z^* \leftarrow\$ \mathcal{A}^{\mathrm{Ver}}(pk, (R_1, c_1, z_1), \ldots, (R_{q_{id}}, c_{q_{id}}, z_{q_{id}}))$
Ret $V.ver(pk, R^*, c^*, z^*)$

$\underline{\mathrm{Ver}(R^*)}$
If queried = true then Ret $\perp$
queried $\leftarrow$ true
$c^* \leftarrow\$ \mathcal{C}$
Ret c^*

$$\mathbf{Adv}^{\mathrm{idpass}}_{\mathsf{ID},q_{id}}(\mathcal{A}) = \Pr\left[\,\mathrm{IDPASS}^{\mathcal{A}}_{\mathsf{ID},q_{id}} \Rightarrow \mathsf{true}\,\right]$$

Schnorr is honest-verifier zero-knowledge

How can we create believable transcripts without secret key for pk = X = g^x

Real transcript:	Faked transcript:
g^r	$R = g^z X^{-c}$
c	c
$z = cx + r$	z

Intuition for security:

1. Can simulate R_i , c_i , z_i because Schnorr is zero-knowledge

2. Build DL-adversary *B* that runs *A* twice to get two transcripts that allow extracting sk = x

$\underline{\mathrm{IDPASS}^{\mathcal{A}}_{\mathsf{ID},q_{id}}}$
$(pk, sk) \leftarrow\$ \mathsf{IDkg}$
$\mathsf{queried} \leftarrow \mathsf{false}$
For $i = 1$ to q_{id} do
  $R_i \leftarrow\$ P.com(pk)$
  $c_i \leftarrow\$ \mathcal{C}$
  $z_i \leftarrow\$ P.resp(sk, R_i, c_i)$
$z^* \leftarrow\$ \mathcal{A}^{\mathrm{Ver}}(pk, (R_1, c_1, z_1), \ldots, (R_{q_{id}}, c_{q_{id}}, z_{q_{id}}))$
Ret $V.ver(pk, R^*, c^*, z^*)$

$\underline{\mathrm{Ver}(R^*)}$
If $\mathsf{queried} = \mathsf{true}$ then Ret $\perp$
$\mathsf{queried} \leftarrow \mathsf{true}$
$c^* \leftarrow\$ \mathcal{C}$
Ret c^*

$$\mathbf{Adv}^{\mathrm{idpass}}_{\mathsf{ID},q_{id}}(\mathcal{A}) = \Pr\left[\, \mathrm{IDPASS}^{\mathcal{A}}_{\mathsf{ID},q_{id}} \Rightarrow \mathsf{true} \,\right]$$

Rewinding trick

Prover given pk and (faked) transcripts of q_{id} honest executions

R*

c*

z*

DL adversary: Given X, find x s.t. $g^x = X$

$R^* X^{c^*} = g^{z^*}$

$r^* + xc^* = z^*$

Prover given ***same*** pk and (faked) transcripts of q_{id} honest executions

R*

c**

z**

$R^* X^{c^{**}} = g^{z^{**}}$

$r^* + xc^{**} = z^{**}$

$x = (z^* - z^{**}) / (c^* - c^{**})$

$\underline{\mathrm{IDPASS}^{\mathcal{A}}_{\mathsf{ID},q_{id}}}$

$(pk, sk) \leftarrow\!\!\$\ \mathsf{IDkg}$

queried $\leftarrow$ false

For $i = 1$ to q_{id} do

$R_i \leftarrow\!\!\$\ P.com(pk)$

$c_i \leftarrow\!\!\$\ \mathcal{C}$

$z_i \leftarrow\!\!\$\ P.resp(sk, R_i, c_i)$

$z^* \leftarrow\!\!\$\ \mathcal{A}^{\mathrm{Ver}}(pk, (R_1, c_1, z_1), \ldots, (R_{q_{id}}, c_{q_{id}}, z_{q_{id}}))$

Ret $V.ver(pk, R^*, c^*, z^*)$

$\underline{\mathrm{Ver}(R^*)}$

If queried = true then Ret $\perp$

queried $\leftarrow$ true

$c^* \leftarrow\!\!\$\ \mathcal{C}$

Ret c^*

$$\mathbf{Adv}^{\mathrm{idpass}}_{\mathsf{ID},q_{id}}(\mathcal{A}) = \Pr\left[\ \mathrm{IDPASS}^{\mathcal{A}}_{\mathsf{ID},q_{id}} \Rightarrow \mathsf{true}\ \right]$$

Reduction from IDPASS to DL

Prover given pk and (faked) transcripts of q_{id} honest executions

R*

c*

z*

$R^* X^{c^*} = g^{z^*}$

DL adversary: Given X, find x s.t. $g^x = X$

$r^* + xc^* = z^*$

Prover given ***same*** pk and (faked) transcripts of q_{id} honest executions

R*

c**

z**

$R^* X^{c^{**}} = g^{z^{**}}$

$r^* + xc^{**} = z^{**}$

$x = (z^* - z^{**}) / (c^* - c^{**})$

$\underline{\mathcal{B}(X)}$

queried $\leftarrow$ false

For $i = 1$ to q_{id} do

$z_i \leftarrow\!\!\$\ \mathbb{Z}_q$

$c_i \leftarrow\!\!\$\ \mathbb{Z}_q$

$R_i \leftarrow\!\!\$\ g^{z_i} X^{-c_i}$

$\omega \leftarrow\!\!\$\ \Omega_{\mathcal{A}}$

$z^* \leftarrow \mathcal{A}^{\mathrm{Ver}}(pk, (R_1, c_1, z_1), \ldots, (R_{q_{id}}, c_{q_{id}}, z_{q_{id}})\ ;\ \omega)$

$z^{**} \leftarrow \mathcal{A}^{\mathrm{Ver}}(pk, (R_1, c_1, z_1), \ldots, (R_{q_{id}}, c_{q_{id}}, z_{q_{id}})\ ;\ \omega)$

If $c^* = c^{**}$ then Ret $x \leftarrow\!\!\$\ \mathbb{Z}_q$

$x \leftarrow (z^* - z^{**})/(c^* - c^{**})$

Ret x

$\underline{\mathrm{Ver}(R^*)}$

If queried = false then

queried $\leftarrow$ true

Ret $c^* \leftarrow\!\!\$\ \mathbb{Z}_q$

Ret $c^{**} \leftarrow\!\!\$\ \mathbb{Z}_q$

Reduction from IDPASS to DL

If *A* succeeds with probability ϵ with c*, can we lower bound probability *A* succeeds twice?

Key analysis captured by ***rewinding lemma (aka reset lemma):***

Lemma. *Let S and T be finite, non-empty sets and let $f : S \times T \to \{0,1\}$ be a function. Let $\mathcal{X}, \mathcal{Y}, \mathcal{Y}'$ be independent random variables, with $\mathcal{X}$ taking values in T and $\mathcal{Y}, \mathcal{Y}'$ being uniformly chosen from T. Let $\Pr[f(\mathcal{X}, \mathcal{Y}) = 1] = \epsilon$. Then*

$$\Pr\left[\, f(\mathcal{X}, \mathcal{Y}) = 1 \wedge f(\mathcal{X}, \mathcal{Y}') = 1 \wedge \mathcal{Y} \neq \mathcal{Y}' \,\right] \geq \epsilon^2 - \frac{\epsilon}{|T|}$$

$\underline{\mathcal{B}(X)}$

queried $\leftarrow$ false

For $i = 1$ to q_{id} do

  $z_i \leftarrow\!\!\$\ \mathbb{Z}_q$

  $c_i \leftarrow\!\!\$\ \mathbb{Z}_q$

  $R_i \leftarrow\!\!\$\ g^{z_i} X^{-c_i}$

$\omega \leftarrow\!\!\$\ \Omega_{\mathcal{A}}$

$z^* \leftarrow \mathcal{A}^{\mathrm{Ver}}(pk, (R_1, c_1, z_1), \ldots, (R_{q_{id}}, c_{q_{id}}, z_{q_{id}})\ ;\ \omega)$

$z^{**} \leftarrow \mathcal{A}^{\mathrm{Ver}}(pk, (R_1, c_1, z_1), \ldots, (R_{q_{id}}, c_{q_{id}}, z_{q_{id}})\ ;\ \omega)$

If $c^* = c^{**}$ then Ret $x \leftarrow\!\!\$\ \mathbb{Z}_q$

$x \leftarrow (z^* - z^{**})/(c^* - c^{**})$

Ret x

$\underline{\mathrm{Ver}(R^*)}$

If queried = false then

  queried $\leftarrow$ true

  Ret $c^* \leftarrow\!\!\$\ \mathbb{Z}_q$

Ret $c^{**} \leftarrow\!\!\$\ \mathbb{Z}_q$

IDPASS security of Schnorr

Theorem. *Let G be a cyclic group, $q_{id} \geq 0$, and $\mathcal{A}$ be an* $\mathrm{IDPASS}_{\mathsf{ID},q_{id}}$*-adversary for the Schnorr scheme ID built using G. Then we give an* DL_G*-adversary $\mathcal{B}$ such that*

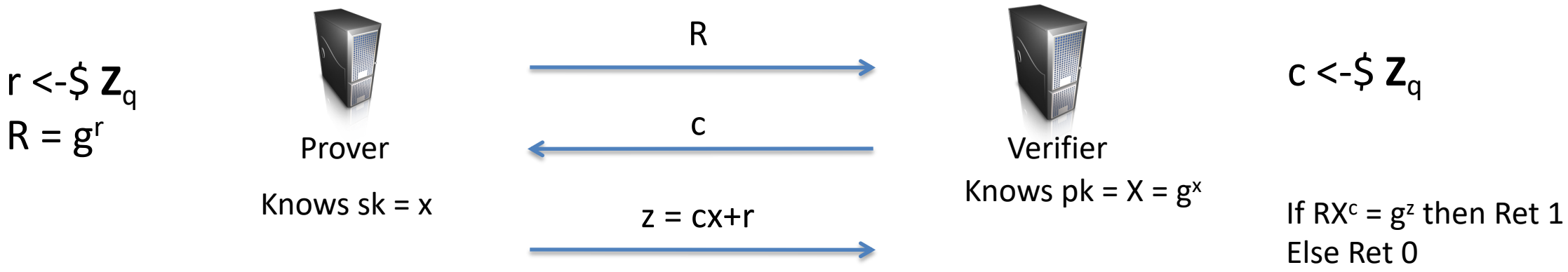
$$\mathbf{Adv}^{\mathrm{idpass}}_{\mathsf{ID},q_{id}}(\mathcal{A}) \leq \sqrt{\mathbf{Adv}^{\mathrm{dl}}_{G}(\mathcal{B})} + \frac{1}{|G|} \ .$$

Adversary $\mathcal{B}$ runs in time at most twice that of $\mathcal{A}$ plus $\mathcal{O}(q_{id})$.

Analysis: game plan

- Formalize ID protocols and their security under ***passive attacks***
- Prove Schnorr ID protocol secure
  - Uses fact that Schnorr protocol is ***honest-verifier zero-knowledge***
  - Requires so-called ***rewinding lemma***
    - Run an adversary twice on related randomness, likely to succeed twice
- Prove that Schnorr signatures UF-CMA implied by Schnorr ID passive attack security

Fiat-Shamir transform



Signature is commitment and response. Challenge generated non-interactively with RO
Works on other Sigma protocols as well

<u>Sign(x, M)</u>
r <-$ Z_q
R = g^r ; c = H(M || R) ; z = r + cx mod q
Return (R,z)

<u>Ver(X, M, (R,z))</u>
c = H(M || R)
If $g^z = RX^c$ then Return 1
Return 0

UF-CMA to IDPASS reduction

Idea:

IDPASS adversary *B* that runs *A:*

- guess RO query i* corresponds to forgery, query ver to set $H[M_{i^*}] = c^*$
- use c_i values as response to other RO queries
- use (R_i, c_i, z_i) to simulate signing queries

Simulates *A*'s environment perfectly if i* guess correct and no commitments R collide with previous H queries

Sign(x, M)

r <-$ $\mathbf{Z}_q$

$R = g^r$; $c = H(M \,||\, R)$; $z = r + cx \mod q$

Return (R,z)

$\underline{\text{IDPASS}^{\mathcal{A}}_{\text{ID},q_{id}}}$

$(pk, sk) \leftarrow\$ \text{IDkg}$

queried $\leftarrow$ false

For $i = 1$ to q_{id} do

$R_i \leftarrow\$ P.com(pk)$

$c_i \leftarrow\$ \mathcal{C}$

$z_i \leftarrow\$ P.resp(sk, R_i, c_i)$

$z^* \leftarrow\$ \mathcal{A}^{\text{Ver}}(pk, (R_1, c_1, z_1), \ldots, (R_{q_{id}}, c_{q_{id}}, z_{q_{id}}))$

Ret $V.ver(pk, R^*, c^*, z^*)$

$\underline{\text{Ver}(R^*)}$

If queried = true then Ret $\perp$

queried $\leftarrow$ true

$c^* \leftarrow\$ \mathcal{C}$

Ret c^*

Putting it all together: UF-CMA to DL

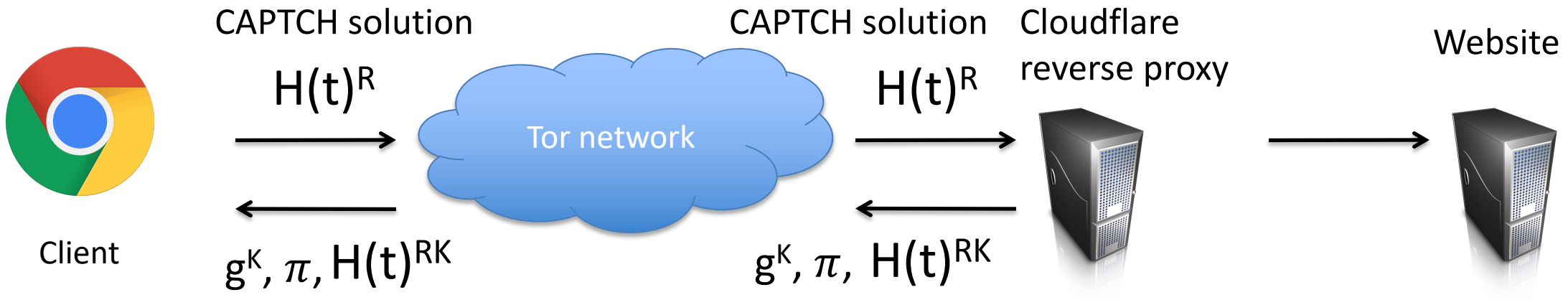
Theorem. *Let G be a cyclic group and* DS *be the Schnorr digital signature scheme using random oracle* $\mathsf{H}\colon \mathcal{M} \to \mathbb{Z}_{|G|}$. *Let $\mathcal{A}$ be an* $\text{UF-CMA}_{\mathsf{DS}}$*-adversary making at most q_h RO queries and q_s signing queries. Then we give an* DL_G*-adversary $\mathcal{B}$ such that*

$$\mathbf{Adv}^{\text{uf-cma}}_{\mathsf{DS}}(\mathcal{A}) \leq \frac{q_s(q_s+q_h+1)}{|G|} + (q_h+1)\left(\sqrt{\mathbf{Adv}^{\text{dl}}_{G}(\mathcal{B})} + \frac{1}{|G|}\right) .$$

Adversary $\mathcal{B}$ runs in time at most twice that of the sum of the running time of $\mathcal{A}$ and $\mathcal{O}(q_h + q_s)$.

Similar type of analysis can be used to show soundness of Fiat-Shamir NIZK for Chaum-Pedersen Sigma protocol

Privacy Pass



$t \leftarrow\$ \{0,1\}^n$
$R \leftarrow\$ \mathbf{Z}_p$
$T = H(H(t)^K)$

Verify π
- If fails, then abort
- Otherwise can use tokens

This is called a ***verifiable OPRF***. In words, it allows ensuring consistency (relative to public key g^K) of server behavior

π is a Chaum-Pedersen NIZK of DL equality
- Soundness: Client knows that same K is used, rules out tagging attacks
- Zero knowledge: Client can't use π to learn something about K

NIZKs for General NP Languages

- Can we build NIZKs for arbitrary statements?
  - Interactive proofs for NP long known [Goldreich, Micali, Wigderson 1991]
    - Protocol for proving knowledge of 3-coloring of graph
- CRS-based NIZKs [Groth, Ostrovsky Sahai 2006]
  - Convert statement to circuit C
  - Bilinear-pairing based NIZK whose proofs are size $O(|C|)$
- Groth 2010
  - Show how to do bilinear-pairing based scheme with constant size proofs.
  - Uses knowledge-of-exponent assumptions
- zkSNARKs
  - More efficient representation than circuits: quadratic span programs
  - Then use Groth-style bilinear pairing proofs
  - Pinocchio system: 288 byte proofs of arbitrary programs

Emerging topics in applied crypto

- We'll spend a couple lectures covering some cryptographic tools that have started seeing use in practice recently
  - Secure computation
  - Oblivious PRFs
  - Zero-knowledge proofs

- Theory goes back decades, but real-world applications were historically elusive