One-Shot Verifiable Encryption from Lattices

Vadim Lyubashevsky and Gregory Neven

IBM Research -- Zurich

Zero-Knowledge Proofs

Zero-Knowledge Proofs

Relation $f(s)=t$, and want to prove knowledge of s

Zero-Knowledge Proofs

Relation $f(s)=t$, and want to prove knowledge of s

e.g. discrete log: Prove knowledge of s s.t. $g^s=t$

Zero-Knowledge Proofs

Relation f(s)=t, and want to prove knowledge of s

e.g. discrete log: Prove knowledge of s s.t. $g^s=t$

For lattice problems such as SIS and LWE,

want to prove knowledge of a **short** vector s

such that f(s)=t

Examples

SIS Problem:

$f_A(s) := As \bmod q$

$$\begin{bmatrix} 4 & 11 & 6 & 8 & 10 & 7 & 6 & 14 \\ 7 & 7 & 1 & 2 & 13 & 0 & 3 & 0 \\ 2 & 9 & 12 & 5 & 1 & 2 & 5 & 9 \\ 1 & 3 & 14 & 9 & 7 & 1 & 11 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 8 \\ 12 \\ 14 \\ 5 \end{bmatrix} \bmod 17$$

Examples

LWE Problem:

$f_A(s) := As \bmod q$

$$\begin{bmatrix} 4 & 11 & 6 & 8 & 1 & 0 & 0 & 0 \\ 7 & 7 & 1 & 2 & 0 & 1 & 0 & 0 \\ 2 & 9 & 12 & 5 & 0 & 0 & 1 & 0 \\ 1 & 3 & 14 & 9 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 12 \\ 10 \\ 8 \\ 10 \end{bmatrix} \mod 17$$

Polynomial Rings

$R = Z_q[x]/(x^d+1)$ is a polynomial ring with

- Addition mod q
- Polynomial multiplication mod q and x^d+1

Polynomial Rings

$R = Z_q[x]/(x^d+1)$ is a polynomial ring with

- Addition mod q
- Polynomial multiplication mod q and x^d+1

SIS Problem over R:

$f_A(s) := As \bmod q$

$$\begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ a_6 & a_7 & a_8 & a_9 & a_{10} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

Constructing Zero-Knowledge Proofs

- For discrete log relations – a simple sigma protocol (i.e. Schnorr proof).
  - Can be made non-interactive via the Fiat-Shamir transformation

- For lattice schemes – the main obstacle is that the secret has small length.

"Fiat-Shamir with Aborts" [Lyu '09]

“Fiat-Shamir with Aborts” [Lyu ‘09]

Relation: $f(s)=t$

“Fiat-Shamir with Aborts” [Lyu ‘09]

Relation: f(s)=t

y $\leftarrow$ D

w=f(y)

"Fiat-Shamir with Aborts" [Lyu '09]

Relation: $f(s)=t$

$y \leftarrow D$

$w=f(y)$

$\xrightarrow{\quad w \quad}$

“Fiat-Shamir with Aborts” [Lyu ‘09]

Relation: f(s)=t

y ← D

w=f(y)

$\xrightarrow{\quad w \quad}$

$\xleftarrow{\quad c \quad}$

"Fiat-Shamir with Aborts" [Lyu '09]

Relation: $f(s)=t$

$y \leftarrow D$

$w=f(y)$

$\xrightarrow{\quad w \quad}$

$\xleftarrow{\quad c \quad}$

$z=sc+y$

(Rejection Sample)

$\xrightarrow{\quad z \quad}$

"Fiat-Shamir with Aborts" [Lyu '09]

Relation: f(s)=t

y ← D

w=f(y)

$\xrightarrow{w}$

$\xleftarrow{c}$

z=sc+y

(Rejection Sample)

$\xrightarrow{z}$

||z|| is small and

f(z)=tc+w

“Fiat-Shamir with Aborts” [Lyu ‘09]

Relation: f(s)=t

y ← D

w=f(y)

w

c

c’

z=sc+y

(Rejection Sample)

z

z’

||z|| is small and
f(z)=tc+w

||z’|| is small and
f(z’)=tc’+w

“Fiat-Shamir with Aborts” [Lyu ‘09]

Relation: $f(s)=t$

$y \leftarrow D$

$w=f(y)$

$\xrightarrow{w}$

$\xleftarrow{c}$ $\xleftarrow{c'}$

$z=sc+y$

(Rejection Sample)

$\xrightarrow{z}$ $\xrightarrow{z'}$

$||z||$ is small and $f(z)=tc+w$

$||z'||$ is small and $f(z')=tc'+w$

$f(z-z')=t(c-c')$

"Fiat-Shamir with Aborts" [Lyu '09]

Relation: $f(s)=t$

$y \leftarrow D$

$w=f(y)$

$\xrightarrow{w}$

$\xleftarrow{c}$ $\qquad\qquad$ $\xleftarrow{c'}$

In a non-interactive protocols, replaced with $c=H(w,t)$

$z=sc+y$

(Rejection Sample)

$\xrightarrow{z}$ $\qquad\qquad$ $\xrightarrow{z'}$

$||z||$ is small and $f(z)=tc+w$

$||z'||$ is small and $f(z')=tc'+w$

$f(z-z')=t(c-c')$

Implications of the Extraction

Implications of the Extraction

$$f(z-z')=t(c-c')$$

$$\Big\Downarrow \quad \text{if } (c-c')^{-1} \text{ exists}$$

$$f((z-z')/(c-c'))=t$$

Implications of the Extraction

$$f(z-z')=t(c-c')$$

$$\Big\Downarrow \quad \text{if } (c-c')^{-1} \text{ exists}$$

$$f((z-z')/(c-c'))=t$$

But $(z-z')/(c-c')$ does not necessarily have small coefficients!

Unless … c,c' in $\{0,1\}$ …

But then soundness is only 1/2.

Practical (< 20KB per proof) Applications

Practical (< 20KB per proof) Applications

$f(\hat{s}) = t\hat{c}$

Practical (< 20KB per proof) Applications

$f(\hat{s}) = t\hat{c}$

Domain of c is large

Digital signatures [Lyu ‘09,…], ZK proofs of commitments [BKLP ‘16], (maybe others)

Practical (< 20KB per proof) Applications

$f(\hat{s}) = t\hat{c}$

Domain of c is large

Digital signatures [Lyu '09,...], ZK proofs of commitments [BKLP '16], (maybe others)

Domain of $\hat{c} = \{-1,0,1\}$

$f(\hat{s})=t$ when simultaneously proving many (>> 10,000) relations [Lyu ' 09] + [BDLN '16] + [CDXY '17]

(Stern-type Lattice ZK Proofs)

- Combinatorial based on the code-based Stern identification scheme with 0/1 secrets [Ste '93]
- Can be adapted to larger secrets at a significant efficiency loss [LNSW '13]

(Stern-type Lattice ZK Proofs)

- Combinatorial based on the code-based Stern identification scheme with 0/1 secrets [Ste ‘93]
- Can be adapted to larger secrets at a significant efficiency loss [LNSW ‘13]

- Proofs are almost always >> 1 MB (depending on how big the coefficients of s are)

- Not considered relevant for practical applications

Main Open Problems

$f(\hat{s}) = t\hat{c}$

Domain of c is large

Digital signatures [Lyu '09,...], ZK proofs of commitments [BKLP '16], (maybe others)

Domain of $\hat{c}$ = {-1,0,1}

$f(\hat{s})=t$ when simultaneously proving many (>> 10,000) relations [Lyu ' 09] + [BDLN '16] + [CDXY '17]

Main Open Problems

$f(\hat{s}) = t\hat{c}$

Domain of c is large

Digital signatures [Lyu '09,…], ZK proofs of commitments [BKLP '16], (maybe others)

More applications

Domain of $\hat{c} = \{-1,0,1\}$

$f(\hat{s})=t$ when simultaneously proving many (>> 10,000) relations [Lyu ' 09] + [BDLN '16] + [CDXY '17]

Main Open Problems

$f(\hat{s}) = t\hat{c}$

Domain of c is large

Digital signatures [Lyu '09,…], ZK proofs of commitments [BKLP '16], (maybe others)

More applications

Domain of $\hat{c} = \{-1,0,1\}$

$f(\hat{s})=t$ when simultaneously proving many (>> 10,000) relations [Lyu ' 09] + [BDLN '16] + [CDXY '17]

Decrease the number of required samples

ZK Proof of Plaintext Knowledge and Verifiable Encryption

ZK Proof of Plaintext Knowledge and Verifiable Encryption

Mediating Authority

Sender

Receiver

ZK Proof of Plaintext Knowledge and Verifiable Encryption

Mediating Authority

Publishes pk to some encryption scheme

Have some secret witness w that x is in language L

Sender

Receiver

ZK Proof of Plaintext Knowledge and Verifiable Encryption

Mediating Authority

Publishes pk to some encryption scheme

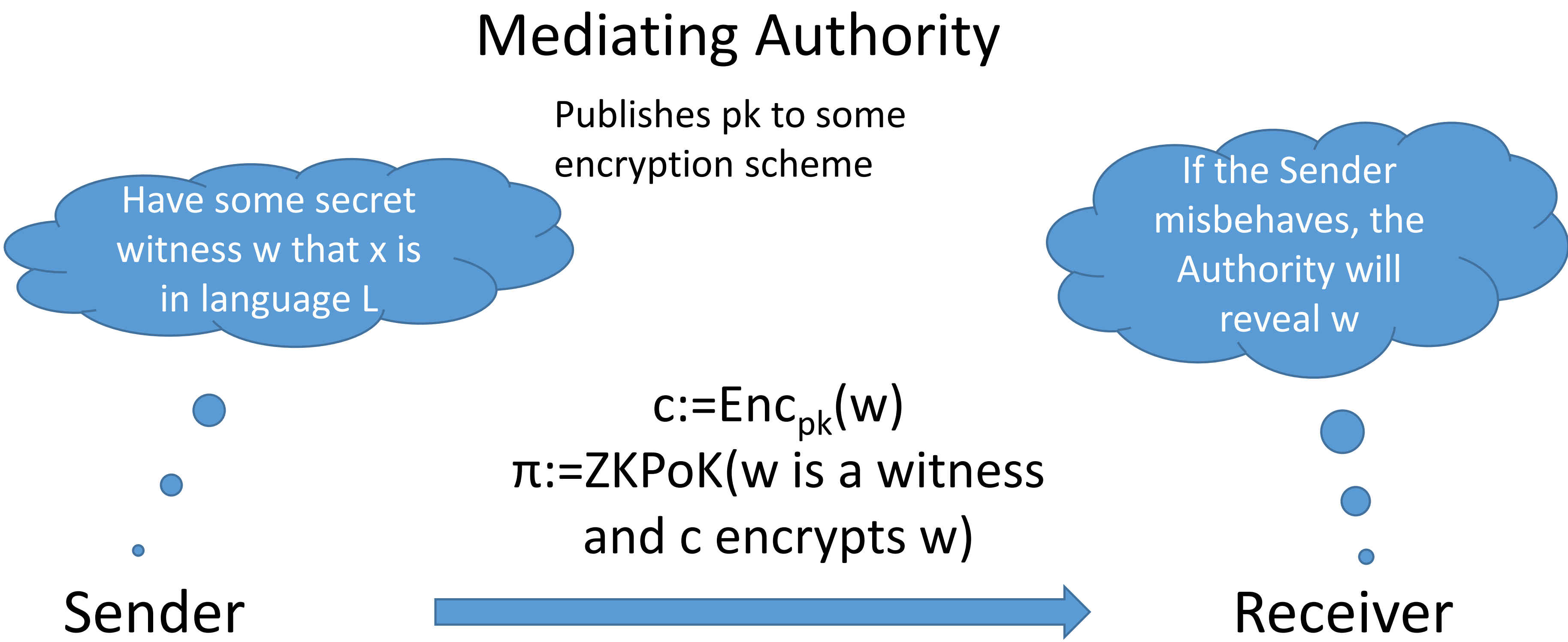
Have some secret witness w that x is in language L

$c := Enc_{pk}(w)$

π:=ZKPoK(w is a witness and c encrypts w)

Sender → Receiver

ZK Proof of Plaintext Knowledge and Verifiable Encryption



ZK Proof of Plaintext Knowledge

Mediating Authority

Publishes pk to some encryption scheme

Have some secret w

If the Sender misbehaves, the Authority will reveal w

$c := Enc_{pk}(w)$

$\pi := ZKPoK(c \text{ encrypts } w)$

Sender → Receiver

Ring-LWE Encryption Scheme

Public Key: a, $as+e=t$

Encryption(m): $u=p(ar+e_1)$, $v=p(tr+e_2)+m$

$$\begin{bmatrix} pa & p & 0 & 0 \\ pt & 0 & p & 1 \end{bmatrix} \begin{bmatrix} r \\ e_1 \\ e_2 \\ w \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix}$$

Decryption: $v-us \bmod q \bmod p$

Approximate Proofs and Proofs of Plaintext Knowledge

$$\begin{bmatrix} pa & p & 0 & 0 \\ pt & 0 & p & 1 \end{bmatrix} \begin{bmatrix} r \\ e_1 \\ e_2 \\ w \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix}$$

Approximate Proofs and Proofs of Plaintext Knowledge

$$\begin{bmatrix} pa & p & 0 & 0 \\ pt & 0 & p & 1 \end{bmatrix} \begin{bmatrix} r \\ e_1 \\ e_2 \\ w \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix}$$

$$\begin{bmatrix} pa & p & 0 & 0 \\ pt & 0 & p & 1 \end{bmatrix} \begin{bmatrix} \hat{r} \\ \hat{e}_1 \\ \hat{e}_2 \\ \hat{w} \end{bmatrix} = \begin{bmatrix} u\hat{c} \\ v\hat{c} \end{bmatrix}$$

Problem with Approximate Proofs

$$\begin{bmatrix} pa & p & 0 & 0 \\ pt & 0 & p & 1 \end{bmatrix} \begin{bmatrix} \hat{r} \\ \hat{e}_1 \\ \hat{e}_2 \\ \hat{w} \end{bmatrix} = \begin{bmatrix} u\hat{c} \\ v\hat{c} \end{bmatrix}$$

Implication: $(v - us)\,\hat{c} \bmod q \bmod p = \hat{w}$

Problem with Approximate Proofs

$$\begin{bmatrix} pa & p & 0 & 0 \\ pt & 0 & p & 1 \end{bmatrix} \begin{bmatrix} \hat{r} \\ \hat{e}_1 \\ \hat{e}_2 \\ \hat{w} \end{bmatrix} = \begin{bmatrix} u\hat{c} \\ v\hat{c} \end{bmatrix}$$

Implication: $(v - us)\, \hat{c} \bmod q \bmod p = \hat{w}$

But decryptor does not know $\hat{c}$

Problem with Approximate Proofs

$$\begin{bmatrix} pa & p & 0 & 0 \\ pt & 0 & p & 1 \end{bmatrix} \begin{bmatrix} \hat{r} \\ \hat{e}_1 \\ \hat{e}_2 \\ \hat{w} \end{bmatrix} = \begin{bmatrix} u\hat{c} \\ v\hat{c} \end{bmatrix}$$

Implication: $(v - us)\,\hat{c} \bmod q \bmod p = \hat{w}$

But decryptor does not know $\hat{c}$

If he decrypts (u,v), he may get garbage because (u,v) is not a valid ciphertext

Our Solution Outline

1. Guess $\hat{c}$

2. $\hat{w}$:=Decrypt $\begin{bmatrix} u\hat{c} \\ v\hat{c} \end{bmatrix}$

3. Output $\hat{w}/\hat{c} \bmod p$

Our Solution Outline

1. Guess $\hat{c}$

There could be $|\text{challenge space}|^2$ possibilities

2. $\hat{w}$:=Decrypt $\begin{bmatrix} u\hat{c} \\ v\hat{c} \end{bmatrix}$

3. Output $\hat{w}/\hat{c} \bmod p$

Our Solution Outline

1. Guess $\hat{c}$
2. $\hat{w}$:=Decrypt $\begin{bmatrix} u\hat{c} \\ v\hat{c} \end{bmatrix}$
3. Output $\hat{w}/\hat{c} \bmod p$

There could be $|\text{challenge space}|^2$ possibilities

How can we be sure we guessed the right $\hat{c}$?

Our Solution Outline

1. Guess $\hat{c}$

There could be $|\text{challenge space}|^2$ possibilities

2. $\hat{w}$:=Decrypt $\boxed{\begin{matrix} u\hat{c} \\ \hline v\hat{c} \end{matrix}}$

How can we be sure we guessed the right $\hat{c}$?

3. Output $\hat{w}/\hat{c}$ mod p

Is this unique? (Decryption should be unique)

1. Guess $\hat{c}$

There could be $|\text{challenge space}|^2$ possibilities

2. $\hat{w}$:=Decrypt $\begin{bmatrix} u\hat{c} \\ v\hat{c} \end{bmatrix}$

How can we be sure we guessed the right $\hat{c}$?

3. Output $\hat{w}/\hat{c}$ mod p

Is this unique? (Decryption should be unique)

1. Guess $\hat{c}$

There could be $|\text{challenge space}|^2$ possibilities

2. $\hat{w}$:=Decrypt $\begin{array}{|c|}\hline u\hat{c} \\ \hline v\hat{c} \\ \hline\end{array}$

How can we be sure we guessed the right $\hat{c}$?

3. Output $\hat{w}/\hat{c}$ mod p

Is this unique? (Decryption should be unique)

We modify the parameters and the decryption algorithm of the Ring-LWE scheme

1. Guess $\hat{c}$

There could be $|\text{challenge space}|^2$ possibilities

2. $\hat{w}$:=Decrypt $\begin{array}{|c|}\hline u\hat{c} \\ \hline v\hat{c} \\ \hline \end{array}$

How can we be sure we guessed the right $\hat{c}$?

3. Output $\hat{w}/\hat{c} \bmod p$

Is this unique? (Decryption should be unique)

We modify the parameters and the decryption algorithm of the Ring-LWE scheme

In the decryption algorithm, check that $\|(v - us)\,\hat{c} \bmod q\|_\infty < q/2C$ where $C=\max \|\hat{c}\|_1$

1. Guess ĉ

There could be |challenge space|^2 possibilities

2. ŵ:=Decrypt $\begin{bmatrix} u\hat{c} \\ v\hat{c} \end{bmatrix}$

How can we be sure we guessed the right ĉ?

3. Output ŵ/ĉ mod p

Is this unique? (Decryption should be unique)

We modify the parameters and the decryption algorithm of the Ring-LWE scheme

In the decryption algorithm, check that $||(v - us)\,\hat{c} \bmod q||_\infty < q/2C$ where $C=\max ||\hat{c}||_1$

For any two ĉ, ĉ' that satisfy the above condition ŵ/ĉ = ŵ'/ĉ' mod p

1. Guess $\hat{c}$

There could be $|\text{challenge space}|^2$ possibilities

1. Guess ĉ

There could be $|\text{challenge space}|^2$ possibilities

If the ciphertext (u,v) is “valid”, then any ĉ (in particular ĉ=1) will lead to a correct decryption

1. Guess ĉ

There could be $|\text{challenge space}|^2$ possibilities

If the ciphertext (u,v) is “valid”, then any ĉ (in particular ĉ=1) will lead to a correct decryption

If the ciphertext (u,v) is “invalid”, then there is some subset of challenges that will allow the adversarial prover to come up with a valid proof

1. Guess ĉ

There could be
|challenge space|^2
possibilities

If the ciphertext (u,v) is “valid”, then any ĉ (in particular ĉ=1) will lead to a correct decryption

If the ciphertext (u,v) is “invalid”, then there is some subset of challenges that will allow the adversarial prover to come up with a valid proof

ĉ = c - c’ where c and c’ are two “successful” challenges
The encryptor / prover already gave one valid proof
So the decryptor already knows one successful challenge

1. Guess ĉ

There could be $|\text{challenge space}|^2$ possibilities

If the ciphertext (u,v) is "valid", then any ĉ (in particular ĉ=1) will lead to a correct decryption

If the ciphertext (u,v) is "invalid", then there is some subset of challenges that will allow the adversarial prover to come up with a valid proof

ĉ = c - c' where c and c' are two "successful" challenges
The encryptor / prover already gave one valid proof
So the decryptor already knows one successful challenge

There could be $|\text{challenge space}|^2$ possibilities

There could be $|\text{challenge space}|^{\cancel{2}}$ possibilities

1. Guess ĉ

There could be |challenge space| possibilities

Theorem:

If a prover is allowed Q queries to the random oracle (where the RO uses coins H), and T is the number of times the decryptor (using coins D) needs to guess ĉ, then:

$$\Pr_{H,D}[T > kQ] < 1/k + \text{negligible}$$

Implications

Implications

Expected decryption time depends on the number of RO queries the adversary makes

Implications

Expected decryption time depends on the number of RO queries the adversary makes

This could be problematic if the adversary is much more powerful than the decryptor

Implications

Expected decryption time depends on the number of RO queries the adversary makes

This could be problematic if the adversary is much more powerful than the decryptor

In many scenarios, the power of the adversary can be mitigated

Limiting the Number of RO Queries by the Adversary

Limiting the Number of RO Queries by the Adversary

1. Make the RO purposefully very slow
   - Honest prover needs 1 RO query
   - Verification only needs 1 RO query
   - Decryption needs 0 RO queries
   - The only entity needing more than 1 is the adversary

Limiting the Number of RO Queries by the Adversary

1. Make the RO purposefully very slow
   - Honest prover needs 1 RO query
   - Verification only needs 1 RO query
   - Decryption needs 0 RO queries
   - The only entity needing more than 1 is the adversary

2. Have an interactive protocol or use public randomness beacons
   - The verifier should send random “salt” to the prover (or the prover should be required to use the public randomness at the time he submits the proof)
   - This restricts pre-computation by the adversary
   - The decryptor is usually off-line, so has more time

Limiting the Number of RO Queries by the Adversary

1. Make the RO purposefully very slow
   - Honest prover needs 1 RO query
   - Verification only needs 1 RO query
   - Decryption needs 0 RO queries
   - The only entity needing more than 1 is the adversary

2. Have an interactive protocol or use public randomness beacons
   - The verifier should send random “salt” to the prover (or the prover should be required to use the public randomness at the time he submits the proof)
   - This restricts pre-computation by the adversary
   - The decryptor is usually off-line, so has more time

3. Impose large fines for cheating
   - The fact that cheating occurred is immediately detected
   - If revealing the cheater’s identity requires decryption, the cheater takes the risk that decryption will succeed

Other Results

Can make the challenge space smaller

- This puts a bound on the maximum number of guesses the decryptor needs to make
- … But increases the proof size

Other Results

Can make the challenge space smaller

- This puts a bound on the maximum number of guesses the decryptor needs to make
- … But increases the proof size

Easy to adapt this to CCA-secure schemes

- Use Naor-Yung approach
- We already have one encryption and a proof, so just add a second encryption

Open Problem

Is this tight?

$$\Pr_{H,D}[T > kQ] < 1/k + \text{negligible}$$

Open Problem

Is this tight?

$$Pr_{H,D}[T > kQ] < 1/k + \text{negligible}$$

Our proof is "black-box". That is, we only use the fact that there is a zero-knowledge proof.

Open Problem

Is this tight?

$$\Pr_{H,D}[T > kQ] < 1/k + \text{negligible}$$

Our proof is "black-box". That is, we only use the fact that there is a zero-knowledge proof.

A non-black-box approach may look at the algebraic properties of R and figure out how the adversary may cheat. Perhaps in some R, it is harder to cheat.

Thanks.