8th Information Security Conference
Singapore, September 2005

An Oblivious Transfer Protocol with Log-Squared Communication

Helger Lipmaa

Cybernetica AS and University of Tartu (Estonia)

`http://www.cs.ut.ee/~helger`

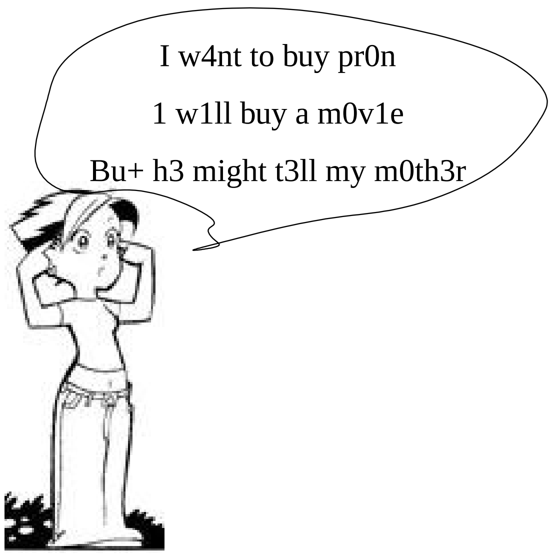
Outline

- Motivation
- Previous Work
- New Construction
- Conclusions

Outline

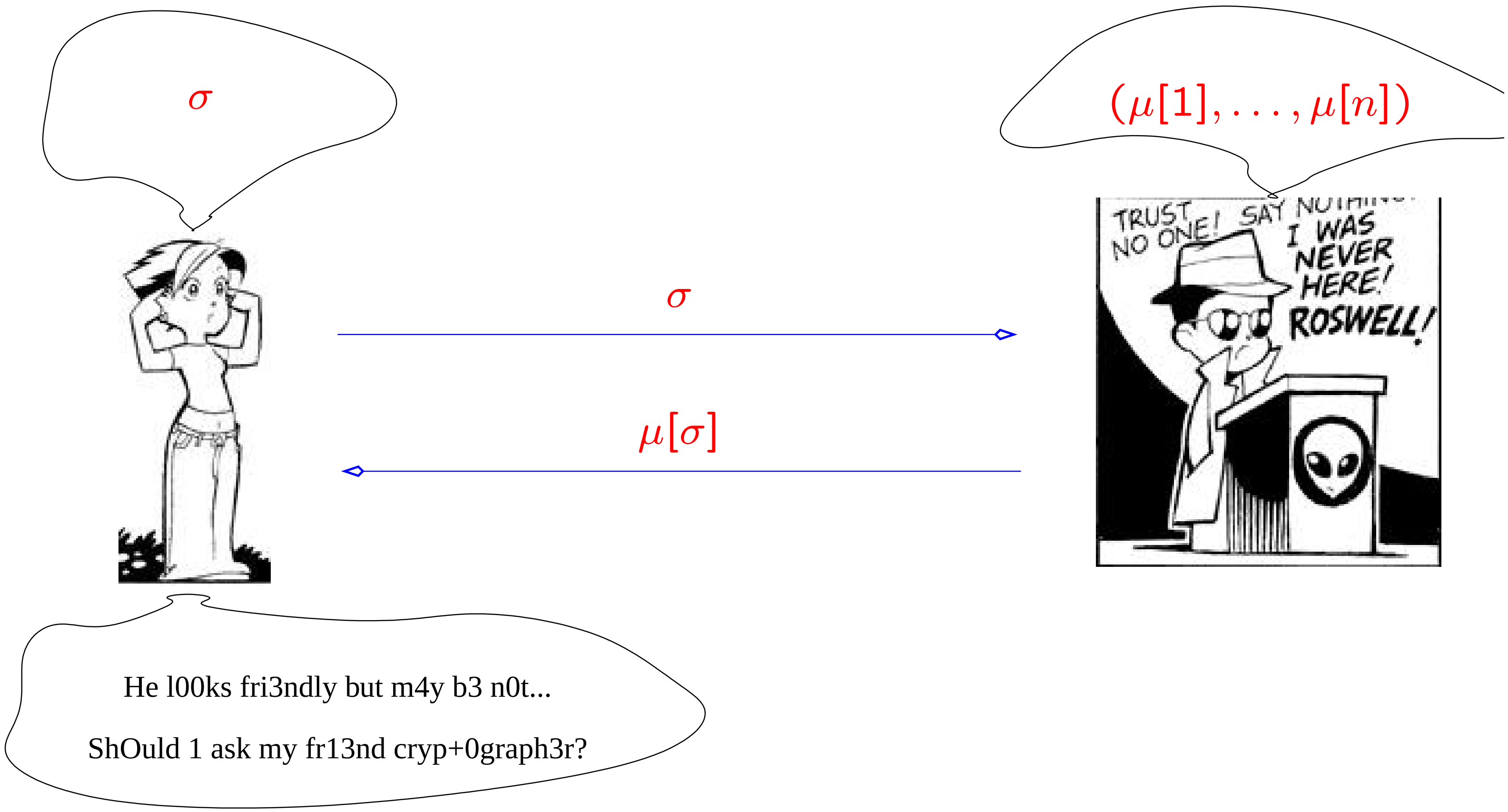
- Motivation
- Previous Work
- New Construction
- Conclusions

Comp.-Private Information Retrieval: Motivation



* *Parental advisory: this is not the only application of PIR–s. Stay tuned!*

Comp.-Private Information Retrieval: Motivation



Comp.-Private Information Retrieval: Motivation

- Chooser wants to retrieve a single element from a database of size n.
  - ⋆ Every element is from $\{0, 1\}^{\ell}$.
- Database maintainer should not know which element was retrieved.
- Security + communication-efficiency.
  - ⋆ Chooser's security is computational (secure if Sender is computationally bounded)
  - ⋆ Information-theoretic security (secure against unbounded Sender): communication is at least $\Omega(n)$.

Comp.-Private Information Retrieval: Motivation

- Non-private version: Monique sends σ, Slick sends $\mu[\sigma]$
  - ⋆ Communication: $\log n + \ell$
- Private version: cannot do better
- Goal: to do as close to $\log n$ as possible
- Intermediate goal:
  - ⋆ Polylogarithmic: $O(\log^b n)$ for some b

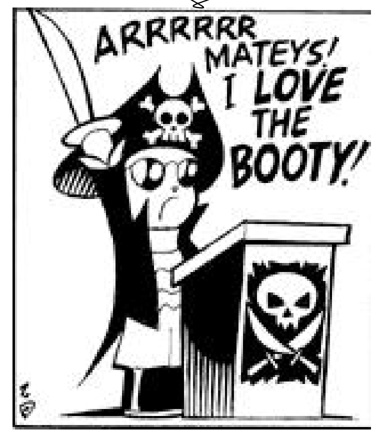
Outline

- Motivation
- Previous Work
- New Construction
- Conclusions

Previous Work

σ

$(\mu[1], \ldots, \mu[n])$



$E_K(\sigma)$

$(E_K(f(\mu[1], \sigma)), \ldots, E_K(f(\mu[n], \sigma)))$

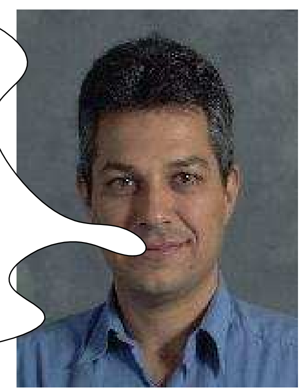
0hn0!
Sl1ck h4s 1,000,000 movies
1ll f1nd an0th3r cryp+3r

Of course you can do that!

It just takes a little work. . .

Like the total length of all movies

. . . times a small constant factor

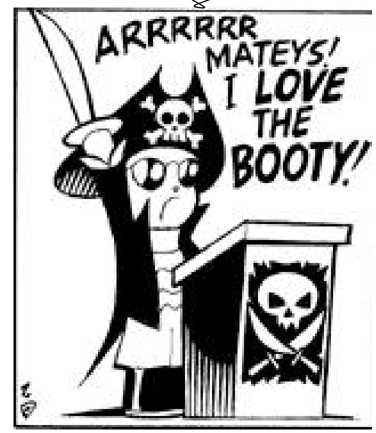


Previous Work

σ

$\{E_K(g_i(\sigma))\}$

$(\mu[1], \ldots, \mu[n])$



$\{E_K(h_i(\mu[\sigma]))\}$

R0xx0rz!
Y0u guys ar3 v3ry cl3v3r!
W4it! 0d3d s4ys th1s is sup3rp07yl0g4r1+hm1c

Of course you can do better!

In fact, better than linear…

You need only $O(2^{\sqrt{\log n}})$ work!

T-this is very s-suprising!

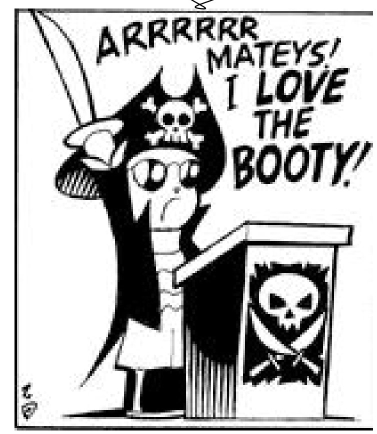
Previous Work

σ

$(\mu[1], \ldots, \mu[n])$

Some gibberish

Some more…gibberish



P07y70g4r1+hm1c!
F1n477y, finally! 1 th4nk y0ur m0th3rs!
W4it! It 1s r34lly n0t pr4ct1c4l?

Polylogarithmicity rocks!

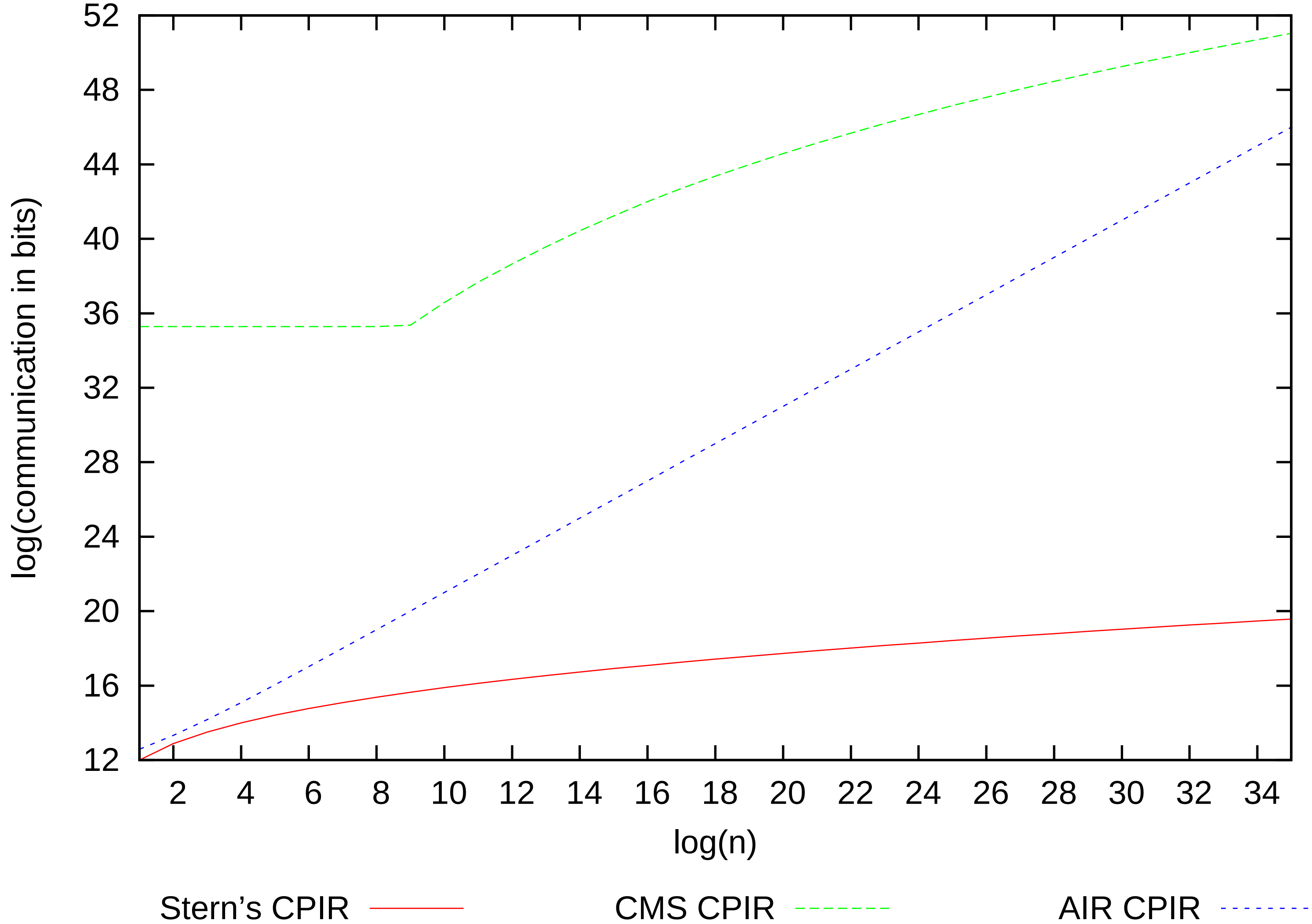
We have this very nice scheme…

You just must hide your Φ

Oded likes it! Practice is for kiddies!

Got very famous!

Previous Work: Overview



Previous Work: Overview

- [Aiello, Ishai, Reingold 2001][Naor, Pinkas, 2001]:
  1-round, $O(\ell \cdot n)$ communication.
  (Protects also the server.)

- [Kushilevitz, Ostrovsky, 1997][Stern, 1998]:
  improved communication to $O(\ell \cdot \sqrt{\log n} \cdot 2^{\sqrt{\log n}})$.

  - $\star$ Not polylogarithmic, but up to now the most practical!

- [Cachin, Micali, Stadler, 1999]: can do polylogarithmic.

  - $\star$ $O(\ell \cdot (\log^8 n + \log^{2f} n))$, $f \geq 4$ unknown (but “constant”!).

- Need: practical and polylogarithmic

Previous Work: Computation

- [Aiello, Ishai, Reingold 2001]:
  - ⋆ Good: Sender's computation $\Theta(n)$
  - ⋆ Good: Client's workload does not depend on n
  - ⋆ Bad: Communication $\Theta(n)$
- [Stern, 1998]:
  - ⋆ Bad: Sender's computation $\Theta(2^{\sqrt{\log n}} \cdot n)$
  - ⋆ Bad: Client's computation $\Theta(\sqrt{\log n} \cdot 2^{\sqrt{\log n}})$
  - ⋆ Good: Communication $\Theta(\sqrt{\log n} \cdot 2^{\sqrt{\log n}})$
- Need: both efficient communication and computation

Outline

- Motivation
- Previous Work
- New Construction
- Conclusions

Generic Idea

- Consider μ as an α-dimensional database, and $\sigma = (\sigma_1, \ldots, \sigma_\alpha)$ as coordinates of the requested element

- Chooser sends encrypted coordinates to Sender

- Sender reduces recursively the dimension of the database by computing intermediate i-dimensional databases of ciphertexts

- The final, 1-dimensional, database is an α-times encryption of requested element. Sender returns it to Chooser

Generic Idea

- Use a length-flexible additively homomorphic public-key cryptosystem.
  - ⋆ $\forall s \geq 1$: encrypts plaintext of sk bits to a ciphertext of $(s+1)k$ bits.
  - ⋆ $E_K^s(m_1)E_K^s(m_2) = E_K^s(m_1 + m_2)$, thus also

$$E_K^{s+1}\Big(\underbrace{m_1}_{(s+1)k}\Big)^{\overbrace{E_K^s(\overbrace{m_2}^{sk})}^{(s+1)k}} = \overbrace{E_K^{s+1}\Big(\underbrace{m_1 E_K^s(m_2)}_{(s+1)k}\Big)}^{(s+2)k} \enspace .$$

- Chooser knows the secret key, Sender knows the public key.
- Sender operates on ciphertexts, sent by Chooser.
- The length parameter s grows in the process.

Generic Idea ($\alpha = 2$)

$\beta_{11} = E_K^s(0)$	$\beta_{12} = E_K^s(0)$	$\beta_{13} = E_K^s(1)$	$\beta_{14} = E_K^s(0)$		
$\mu(1,1)$	$\mu(2,1)$	$\mu(3,1)$	$\mu(4,1)$	$\Rightarrow$	$w_{11} = \prod_i \beta_{1i}^{\mu(1,i)} = E_K^s(\mu(1,\sigma_1))$
$\mu(1,2)$	$\mu(2,2)$	$\mu(3,2)$	$\mu(4,2)$	$\Rightarrow$	$w_{12} = \prod_i \beta_{1i}^{\mu(2,i)} = E_K^s(\mu(2,\sigma_1))$
$\mu(1,3)$	$\mu(2,3)$	$\boxed{\mu(3,3)}$	$\mu(4,3)$	$\Rightarrow$	$\boxed{w_{13} = \prod_i \beta_{1i}^{\mu(3,i)} = E_K^s(\mu(3,\sigma_1))}$
$\mu(1,4)$	$\mu(2,4)$	$\mu(3,4)$	$\mu(4,4)$	$\Rightarrow$	$w_{14} = \prod_i \beta_{1i}^{\mu(1,i)} = E_K^s(\mu(1,\sigma_1))$
sk bits	sk bits	sk bits	sk bits		$(s+1)k$ bits

Generic Idea ($\alpha = 2$)

$\beta_{11} = E^s_K(0)$ $\beta_{12} = E^s_K(0)$ $\beta_{13} = E^s_K(1)$ $\beta_{14} = E^s_K(0)$

Chooser sends $\{\beta_{jt} = E^s_K(\sigma_j \stackrel{?}{=} t)\}$ to Sender

$\mu(1,1)$	$\mu(2,1)$	$\mu(3,1)$	$\mu(4,1)$
$\mu(1,2)$	$\mu(2,2)$	$\mu(3,2)$	$\mu(4,2)$
$\mu(1,3)$	$\mu(2,3)$	$\boxed{\mu(3,3)}$	$\mu(4,3)$
$\mu(1,4)$	$\mu(2,4)$	$\mu(3,4)$	$\mu(4,4)$

$\Rightarrow$ $w_{11} = \prod_i \beta_{1i}^{\mu(1,i)} = E^s_K(\mu(1,\sigma_1))$ — $\beta_{21} = E^{s+1}_K(0)$

$\Rightarrow$ $w_{12} = \prod_i \beta_{1i}^{\mu(2,i)} = E^s_K(\mu(2,\sigma_1))$ — $\beta_{22} = E^{s+1}_K(0)$

$\Rightarrow$ $\boxed{w_{13} = \prod_i \beta_{1i}^{\mu(3,i)} = E^s_K(\mu(3,\sigma_1))}$ — $\beta_{23} = E^{s+1}_K(1)$

$\Rightarrow$ $w_{14} = \prod_i \beta_{1i}^{\mu(1,i)} = E^s_K(\mu(1,\sigma_1))$ — $\beta_{24} = E^{s+1}_K(0)$

$\Downarrow$

$$\boxed{w_2 = \prod_i \beta_{2i}^{w_{1i}} = E^{s+1}_K(E^s_K(\mu(\sigma_1,\sigma_2)))}$$

Chooser sends: $\sum_{j=1}^{\alpha} \sum_{t=1}^{n^{1/\alpha}} (s+j)k$ bits

Sender sends $(s+\alpha)k$ bits

Communication

- Suitable for ℓ-bit strings.

- Chooser sends $\alpha(s + \frac{\alpha+1}{2})n^{1/\alpha}k$ bits.

  ⋆ $sk \approx \ell$, thus $(\ell\alpha + \alpha \cdot \frac{\alpha+1}{2}k)n^{1/\alpha}$ bits.

- Optimal if $\alpha = \Theta(\log n)$: $\Theta(k \cdot \log^2 n + \ell \cdot \log n)$ bits.

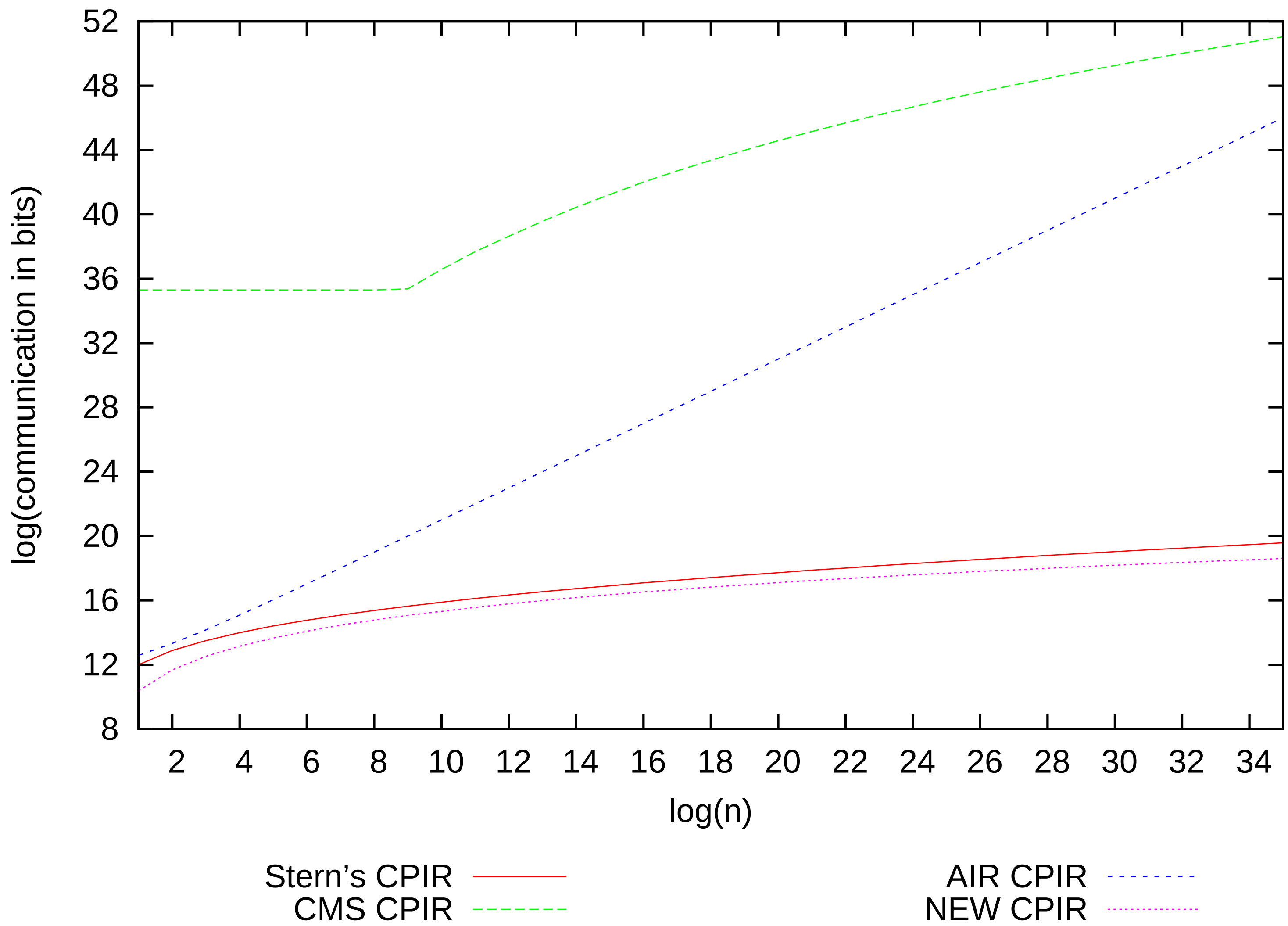
- Very good if $\ell = \mathcal{LARGE}$: $\Theta(\ell \cdot \log n)$ bits.

- Paper discusses various optimisations

  ⋆ For small ℓ, pack several database elements into one plaintext, and assume μ is a lopsided hyperrectangle.

- "Cleaner" and more efficient than previous solutions

Polylogarithmic Yet Practical



Computation

- [Stern, 1998]:
  - ⋆ Bad: Sender's computation $\Theta(2^{\sqrt{\log n}} \cdot n)$
  - ⋆ Bad: Client's computation $\Theta(\sqrt{\log n} \cdot 2^{\sqrt{\log n}})$
  - ⋆ Good: Communication $\Theta(\sqrt{\log n} \cdot 2^{\sqrt{\log n}})$
- New scheme:
  - ⋆ Good: Sender's computation $\Theta(n)$
  - ⋆ Good: Client's computation $\Theta(\log^{2+o(1)} n)$
  - ⋆ Better: Communication $\Theta(\log^2 n)$

Security

- Secure if based on any IND-CPA secure pkc
  - ⋆ Loose reduction
- Secure if based on a new IND-LFCPA assumption
  - ⋆ Tight reduction
- Both existing length-flexible pkc's are tightly IND-LFCPA secure
- Natural assumption!

Stronger Security Notion

- Previous security proofs guarantee security against adversary that works in time τ and has advantage ε

- Sometimes, one wants security against $\mathbf{poly}(n)$-time adversary

- Then $k = \log^{b-o(1)} n$ where the underlying hard problem can be solved in time $\exp(O(1) \cdot (\log n)^{1/b} \cdot (\log \log n)^{1-1/b})$

- With DCRA, $b = 3$, thus our protocol has communication $\Theta(\log^{5-o(1)} n + \ell \cdot \log n)$

Log-Squared Oblivious Transfer

- In CPIR, we care only about Chooser's privacy.

- OT: also Sender's privacy is important.

  - ⋆ Chooser obtains no information about $\mu[i]$ for $i \neq \sigma$.

- To modify the new CPIR into an OT,

  - ⋆ Chooser must prove the correctness of public key. (done once)

  - ⋆ Sender must hide intermediate random values. (easy)

  - ⋆ We must guarantee that Chooser cannot cheat by sending incorrect inputs. (complicated)

Log-Squared Oblivious Transfer: Some Attempts

- [Naor-Pinkas 1999] transformation: with log. overhead in communication, transforms our CPIR to OT.
  - ⋆ Bad: computational server-privacy.

- Zero-knowledge proofs: Chooser proves in ZK that her inputs are correct. Information-theoretical server-privacy.
  - ⋆ Bad: two rounds, or one round but security only in the random-oracle/common reference string model.

Log^2 OT with AIR OT

- [Aiello-Ishai-Reingold]: the AIR CPIR protocol is actually an OT protocol, that can be used in conjunction with any sublinear CPIR protocol to construct an OT protocol with comparable communication.

  ⋆ Chooser only sends one ciphertext to Sender who computes ciphertexts $E_K(\nu[i])$, where $\nu[\sigma] = \mu[\sigma]$ and $\nu[i]$ is "garbage" for $i \neq \sigma$.

  ⋆ In parallel, Chooser executes any CPIR protocol to retrieve $E_K(\nu[\sigma])$.

- In conjunction with the new CPIR, we get an OT protocol with communication $\Theta(k \cdot \log^2 n + \ell \cdot \log n)$.

- Problem: AIR OT is secure only if the DDH holds.

- Thus the resulting log-squared OT is secure only if both the pkc is IND-LFCPA secure and DDH assumption holds.

Log^2 OT with Laur-Lipmaa OT

- [Laur, Lipmaa, manuscript]:
  A similar OT protocol that works over the known length-flexible pkc's.
  - $\star$ Server-privacy is *statistical*
- Results in:
  one-round information-theoretically server-private OT protocol with log-squared communication, secure if assuming that the underlying pkc is IND-LFCPA secure.
- Transformation is very efficient!
- Similar on AIR. . .

Conclusions

- CPIR with log-squared communication: better than "impractical" polylogarithmic CMS CPIR and "practical" superpolylogarithmic CPIR by Stern.

- Security: requires new notion if we want tight security. Purely by luck(?), existing length-flexible pkc's are tightly IND-LFCPA secure.

- Computation: near-optimal.

- Communication: $\Theta(k \cdot \log^2 n + \ell \cdot \log n)$ — note that for large documents, this is $\approx \Theta(\ell \cdot \log n)$.

  ⋆ Non-private information retrieval: $\log n + \ell$ bits — close to optimal!

- Polylogarithmicity is not everything! Exact communication matters.

Any questions?