An Oblivious Transfer Protocol with Log-Squared Communication

Helger Lipmaa
Helsinki University of Technology
`http://www.tcs.hut.fi/~helger`

Outline

- Motivation
- Previous Work
- New Construction
- Conclusions

Outline

- Motivation
- Previous Work
- New Construction
- Conclusions

Comp.-Private Information Retrieval: Motivation

- Chooser wants to retrieve a single element from a database of size n.

- Every element is from $\mathbb{Z}_d$ (with length $\log d$ bits).

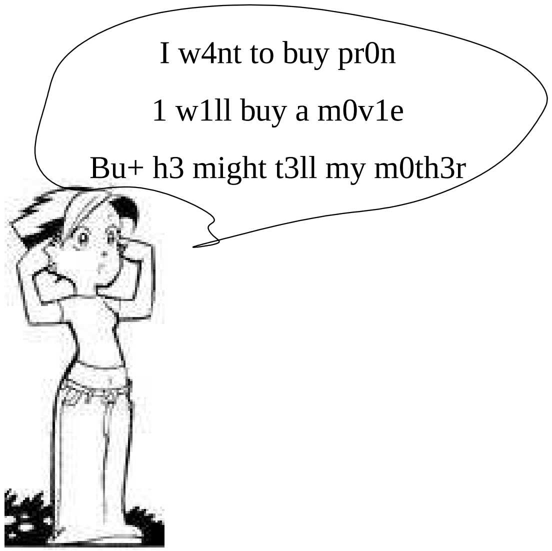
- Database maintainer should not know which element was retrieved.

- Security + communication-efficiency.

  ⋆ Chooser's security is computational.

  ⋆ Information-theoretic security: communication is at least $\Omega(n)$.

Comp.-Private Information Retrieval: Motivation



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

Comp.-Private Information Retrieval: Motivation

σ

σ

$\mu[\sigma]$

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

He l00ks fri3ndly but m4y b3 n0t...

ShOuld 1 ask my fr13nd cryp+0graph3r?

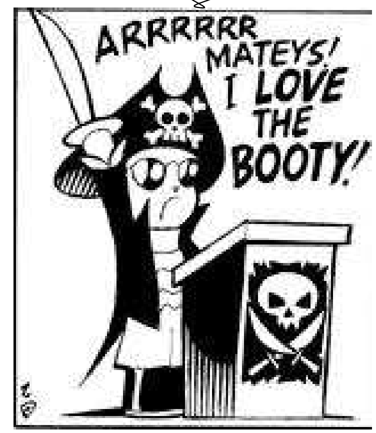
Outline

- Motivation
- Previous Work
- New Construction
- Conclusions

Previous Work

σ

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



$E_K(\sigma)$

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

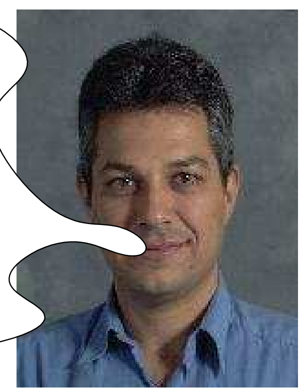
0hn0!
Sl1ck h4s 1,000,000 movies
1ll f1nd an0th3r cryp+3r

Of course you can do that!

It just takes a little communication. . .

Like the total length of all movies

. . . times a small constant factor



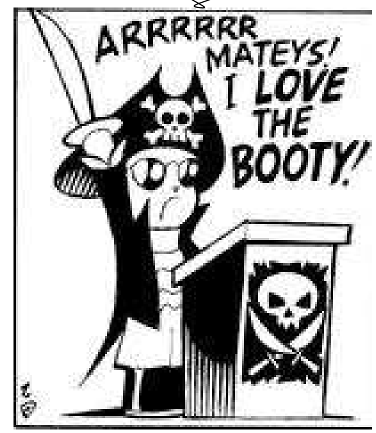
Previous Work

σ

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

$\{E_K(g_i(\sigma))\}$ →

← $\{E_K(h_i(\mu[\sigma]))\}$



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

Of course you can do better!

In fact, better than linear...

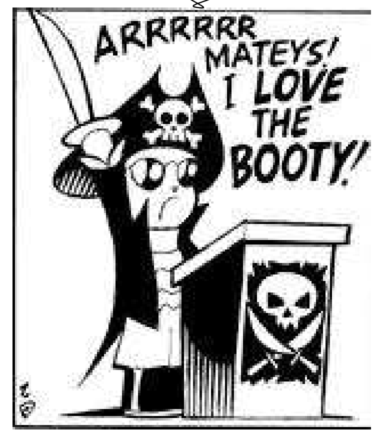
You need only $O(2^{\sqrt{\log n}})$ communication!

T-this is very s-suprising!

Previous Work

σ

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



Some gibberish

Some more. . . gibberish

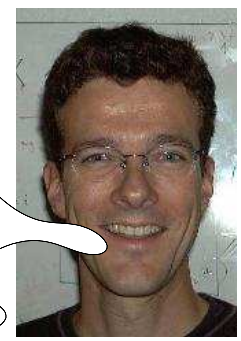
P07y70g4r1+hm1c!
F1n477y, finally! 1 th4nk y0ur m0th3rs!
W4it! It 1s r34lly n0t pr4ct1c4l?

Polylogarithmicity rocks!

We have this very nice scheme. . .

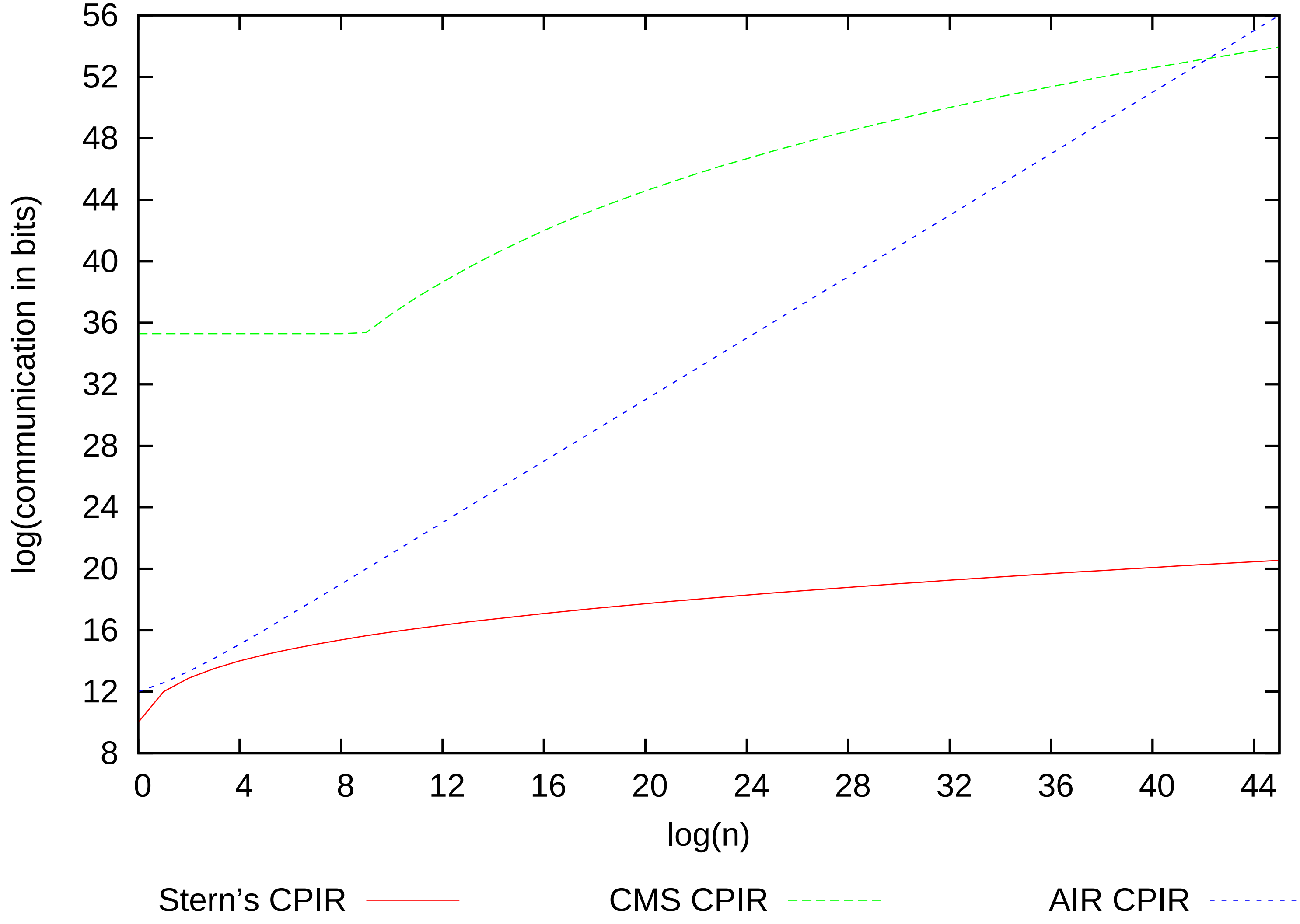
You just must hide your Φ

Oded likes it! Practice is for kiddies!



Got very famous!

Previous Work: Overview



Previous Work: Overview

- [Aiello, Ishai, Reingold 2001][Naor, Pinkas, 2001]: 2-round CPIR, $O(n \cdot \log d)$ communication.

- [Kushilevitz, Ostrovsky, 1997][Stern, 1998][Chang, 2004]: improved communication to $O(\sqrt{\log n} \cdot 2^{\sqrt{\log n}} \cdot \log d)$.

  ⋆ Not polylogarithmic, but up to now the most practical!

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

  ⋆ $O((\log^8 n + \log^{2f} n) \cdot \log d)$, $f \geq 4$ unknown (but "constant"!).

- Need: practical and polylogarithmic

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.
  - $\star$ $\forall s \geq 1$: encrypts plaintext of sk bits to a ciphertext of $(s+1)k$ bits.
  - $\star$ $E_K^s(m_1) E_K^s(m_2) = E_K^s(m_1 + m_2)$, thus also

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

- 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 ($\alpha = 2$)

$\beta_{11} = E_K^s(0)$ $\beta_{12} = E_K^s(0)$ $\beta_{13} = E_K^s(1)$ $\beta_{14} = E_K^s(0)$

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

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

$\Rightarrow$ $w_{11} = \prod_i \beta_{1i}^{\mu(1,i)} = E_K^s(\mu(1,\sigma_1))$ $\qquad \beta_{21} = E_K^{s+1}(0)$

$\Rightarrow$ $w_{12} = \prod_i \beta_{1i}^{\mu(2,i)} = E_K^s(\mu(2,\sigma_1))$ $\qquad \beta_{22} = E_K^{s+1}(0)$

$\Rightarrow$ $\boxed{w_{13} = \prod_i \beta_{1i}^{\mu(3,i)} = E_K^s(\mu(3,\sigma_1))}$ $\qquad \beta_{23} = E_K^{s+1}(1)$

$\Rightarrow$ $w_{14} = \prod_i \beta_{1i}^{\mu(1,i)} = E_K^s(\mu(1,\sigma_1))$ $\qquad \beta_{24} = E_K^{s+1}(0)$

$\Downarrow$

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

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 sending integers from $\mathbb{Z}_d$.

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

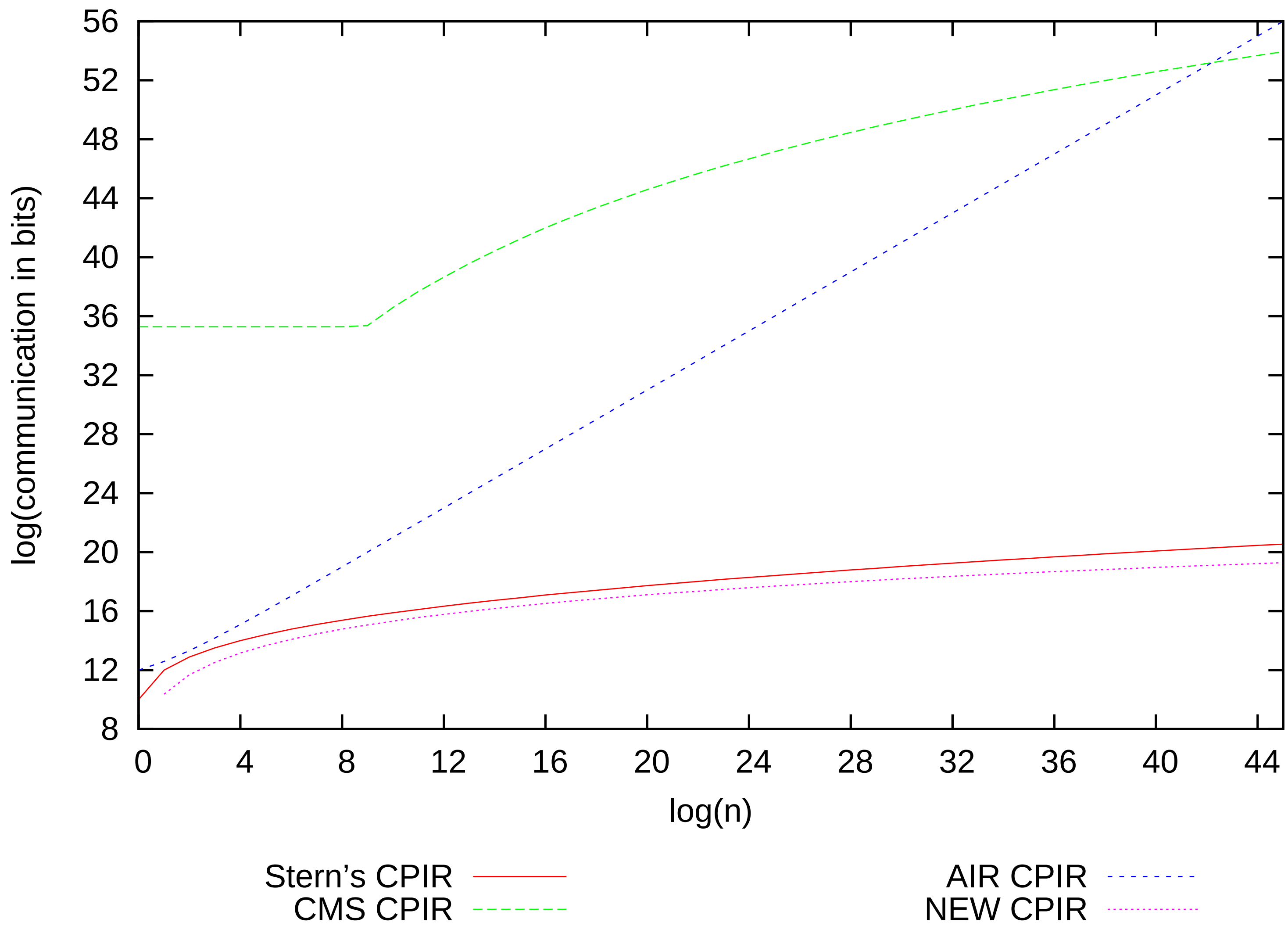
  ⋆ $sk \approx \log d$, thus $(\alpha \log d + \alpha \cdot \frac{\alpha+1}{2}k)n^{1/\alpha}$ bits.

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

- Paper discusses various optimisations

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

Polylogarithmic Yet Practical



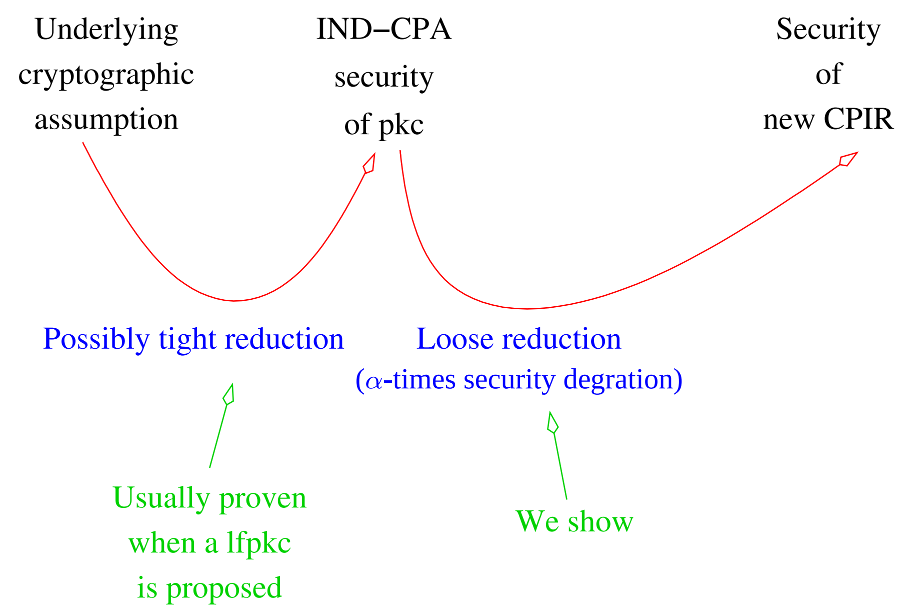
Security: IND-CPA security

- Standard security requirement for homomorphic pkc's: IND-CPA security
  - ⋆ For a randomly chosen key pair, attacker cannot distinguish random encryptions of two plaintexts, chosen by herself.

- We use a *length-flexible* additively homomorphic pkc.

- [Damgård-Jurik 2001, 2003]: There exist IND-CPA secure length-flexible additively homomorphic pkc's.

Security Reduction



- IND-CPA security gives only loose security reduction here
- (Recall that $\alpha = \Theta(\log n)$.)

Why Loose Reduction?

- Length-flexible cryptosystems have been used before to improve the efficiency of e-voting and e-auction schemes.

- There, IND-CPA security gives a tight reduction. Why not here?

- In e-voting/e-auction schemes, the participants send out ciphertexts only with one, fixed, although large, s.

- In our protocol, Chooser sends ciphertexts that correspond to different s's: $\beta_{jt} = E_K^{s+j-1}(\sigma_j =^? t)$.

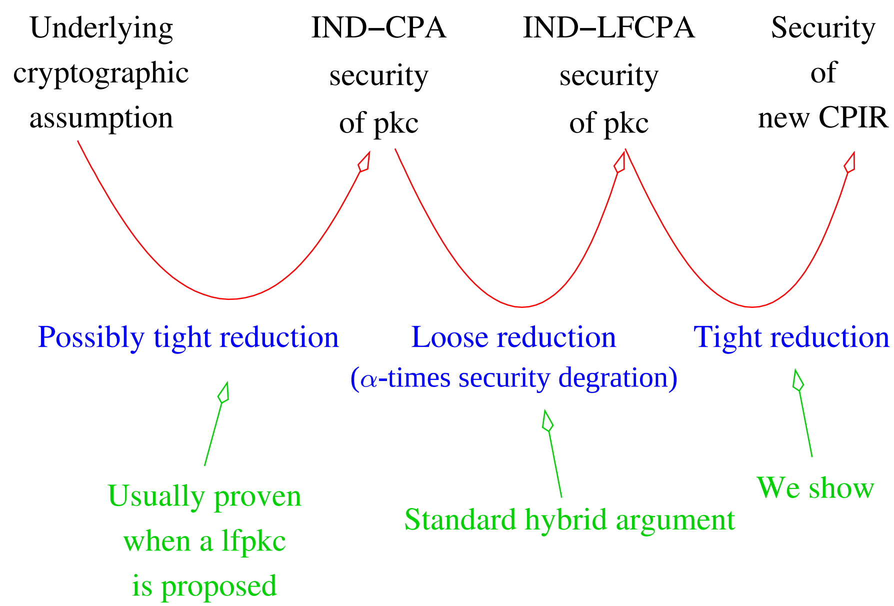
- Thus, the cryptosystem must be secure against attacks where the attacker legally sees ciphertexts of related but unknown plaintexts with different values of s.

New Security Notion: IND-LFCPA Security

Definition A pkc is α-IND-LFCPA secure, if every "efficient" attacker has "small" success in the next game:

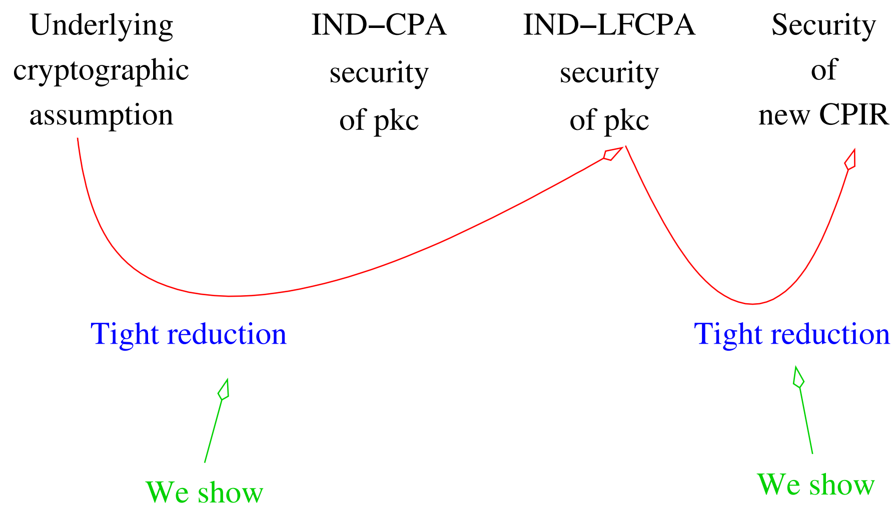
- A random key is chosen, attacker gets the public key.
- Attacker chooses $(m_0, m_1, s_1, \ldots, s_\alpha)$.
- A random $b \leftarrow \{0, 1\}$ is chosen.
- Attacker obtains random ciphertexts $(E_K^{s_1}(m_b), \ldots, E_K^{s_\alpha}(m_b))$.
- Attacker outputs a bit b'.
- Attacker wins if $b = b'$.

Security Reduction: Finer Picture



- Tight reduction to IND-LFCPA security of pkc.
- Loose reduction to IND-CPA security of pkc.
- Loose reduction to underlying cryptographic assumption.

Security Reduction: Damgård-Jurik pkc's



- [DJ '01,'03] pkc's are IND-LFCPA secure with *tight reduction* to DCRA.
- Thus the new CPIR, based on DJ, is secure with tight reduction to DCRA.
- We argue that IND-LFCPA security is such a basic notion that is should be considered standard for length-flexible pkc's.

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: four rounds, or two-rounds 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 $\Theta(\log^2 n \cdot k + \log n \cdot \log d)$.
- 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.

- Result: two-round information-theoretically server-private OT protocol with log-squared communication, secure when assuming that the underlying pkc is IND-LFCPA secure.

- Transformation is very efficient!

Conclusions

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

- Inspired by Stern's CPIR, but uses length-flexible cryptosystems.

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

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

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

- Polylogarithmicity is not everything! Exact communication matters.

Any questions?

Caveat: This presentation is based on a draft version of the paper!

Thanks for inviting!