Advanced Tools from Modern Cryptography

Lecture 17

Homomorphic Encryption. Application to PIR.

Homomorphic Encryption

- Group Homomorphism: Two groups G and G′ are homomorphic if there exists a function (homomorphism) $f:G \to G'$ such that for all $x,y \in G$, $f(x) +_{G'} f(y) = f(x +_G y)$
- Homomorphic Encryption: A CPA secure (public-key) encryption s.t. $\mathrm{Dec}(C) +_M \mathrm{Dec}(D) = \mathrm{Dec}(C +_C D)$ for ciphertexts C, D
  - i.e. $\mathrm{Enc}(x) +_C \mathrm{Enc}(y)$ is like $\mathrm{Enc}(x +_M y)$
  - Interesting when $+_C$ doesn't require the decryption key
- e.g. El Gamal: $(g^{x1},m_1Y^{x1}) \times (g^{x2},m_2Y^{x2}) = (g^{x3},m_1m_2Y^{x3})$

Homomorphic Encryption

- El Gamal needs messages to be in a “hard group” $\mathbb{G}$ (DDH holds)
  - Not a concern in encryption: just use any efficiently computable/invertible mapping from message space $\mathbb{M}$ to $\mathbb{G}$ (efficient inversion needed during decryption)
  - But for homomorphic encryption, group operation will be that of $\mathbb{G}$
    - Since group operation in $\mathbb{M}$ desired, will need mapping from $\mathbb{M}$ to $\mathbb{G}$ to be a homomorphism
    - But if $\mathbb{M}$ is not a hard group (e.g., $\mathbb{Z}_n$), will need $\mathbb{G}$ to have a large enough non-hard subgroup
    - Need a hardness assumption that allows this

Goldwasser-Micali

- Message space $\mathbb{M}$ is $\mathbb{Z}_2$ (i.e., bits with XOR as group operation)
- Ciphertext space contained in $\mathbb{Z}_n^*$ where n = pq, p and q being large primes
- Fact: $(z^{(p-1)/2} \bmod p, z^{(q-1)/2} \bmod q) = (\pm 1, \pm 1)$

$z^{(p-1)} \equiv 1 \bmod p$
$\sqrt{1} = \pm 1 \bmod p$

- Idea: If p, q not given, not easy to find this pair

Jacobi symbol

  - But turns out, can distinguish $\{(+1,+1),(-1,-1)\}$ vs. $\{(+1,-1),(-1,+1)\}$
  - Quadratic Residuosity Assumption: Given only n, hard to distinguish between (+1,+1) and (-1,-1) types
  - Idea: Encryption of 0 is a random z of (+1,+1) type and encryption of 1 is a random z of (-1,-1) type
    - (+1,+1) type can be sampled as $x^2 \bmod n$ for random x
    - (-1,-1) type: Given one z* of (-1,-1) type (can be part of PK), sample $z = x^2.z^*$

Goldwasser-Micali

- Message space $\mathbb{M}$ is $\mathbb{Z}_2$ (i.e., bits with XOR as group operation)
- Public key = (n, z*) where n = pq, p and q being large primes and z* is a random element of type (-1,-1). Secret Key = (p,q).
- Enc: 0 ↦ x^2 mod n, and Enc: 1 ↦ $x^2.z^*$ mod n
- Decryption: Using p, q, compute ($z^{(p-1)/2}$ mod p, $z^{(q-1)/2}$ mod q)
- Homomorphism: Enc(a⊕b) same as Enc(a).Enc(b)

Paillier's Scheme

- Uses $\mathbb{Z}_{n^2}^* \simeq \mathbb{Z}_n \times \mathbb{Z}_n^*$, for a specially chosen n

  n = pq for primes p, q, within 2x of each other

  - Isomorphism: $\psi(a,b) = g^a b^n \pmod{n^2}$ where $g=(1+n)$
- Fact: ψ can be efficiently inverted if factorization of n known
- "Decisional Composite Residuosity" assumption: Given n=pq (but not p,q), ψ(0,rand) looks like ψ(rand,rand) (i.e., random)
- Enc(m) = ψ(m,r) for m in $\mathbb{Z}_n$ and a random r in $\mathbb{Z}_n^*$
- (Additive) Homomorphism: Enc(m).Enc(m') is Enc(m+m')
  - $\psi(m,r).\psi(m',r') = \psi(m+m',r.r')$

in $\mathbb{Z}_{n^2}^*$

in $\mathbb{Z}_n$

- IND-CPA secure under DCR
- Unlinkability: ReRand(c) = c.Enc(0)
- Multiplication by plain-text: $a * \text{Enc}(m) = (\ \psi(m,r)\ )^a = \psi(am,r^a)$

Private Information Retrieval

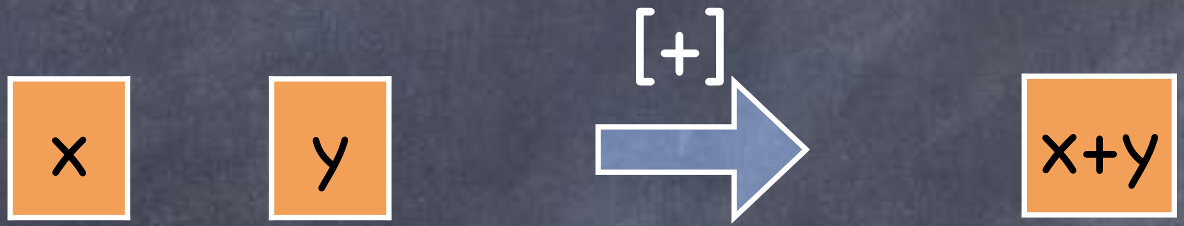
- Setting: A server holds a large vector of values ("database"). Client wants to retrieve the value at a particular index i
  - Client wants privacy against an honest-but-curious server
  - Server has no security requirements
- Trivial solution: Server sends the entire vector to the client
- PIR: to do it with significantly less communication
- Variant (not today): multiple-server PIR, with non-colluding servers

Private Information Retrieval

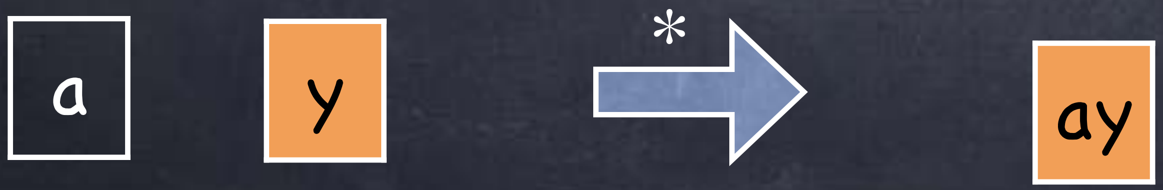
- Single-server PIR using additive homomorphic encryption (need not be unlinkable)
  - Client sends some encrypted representation of the index (need CPA security here)
  - Server operates on the entire database using this encryption (homomorphically), so that the message in the resulting encrypted data has the relevant answer (and maybe more). It sends this (short) encrypted data to client, who decrypts to get answer.

Private Information Retrieval

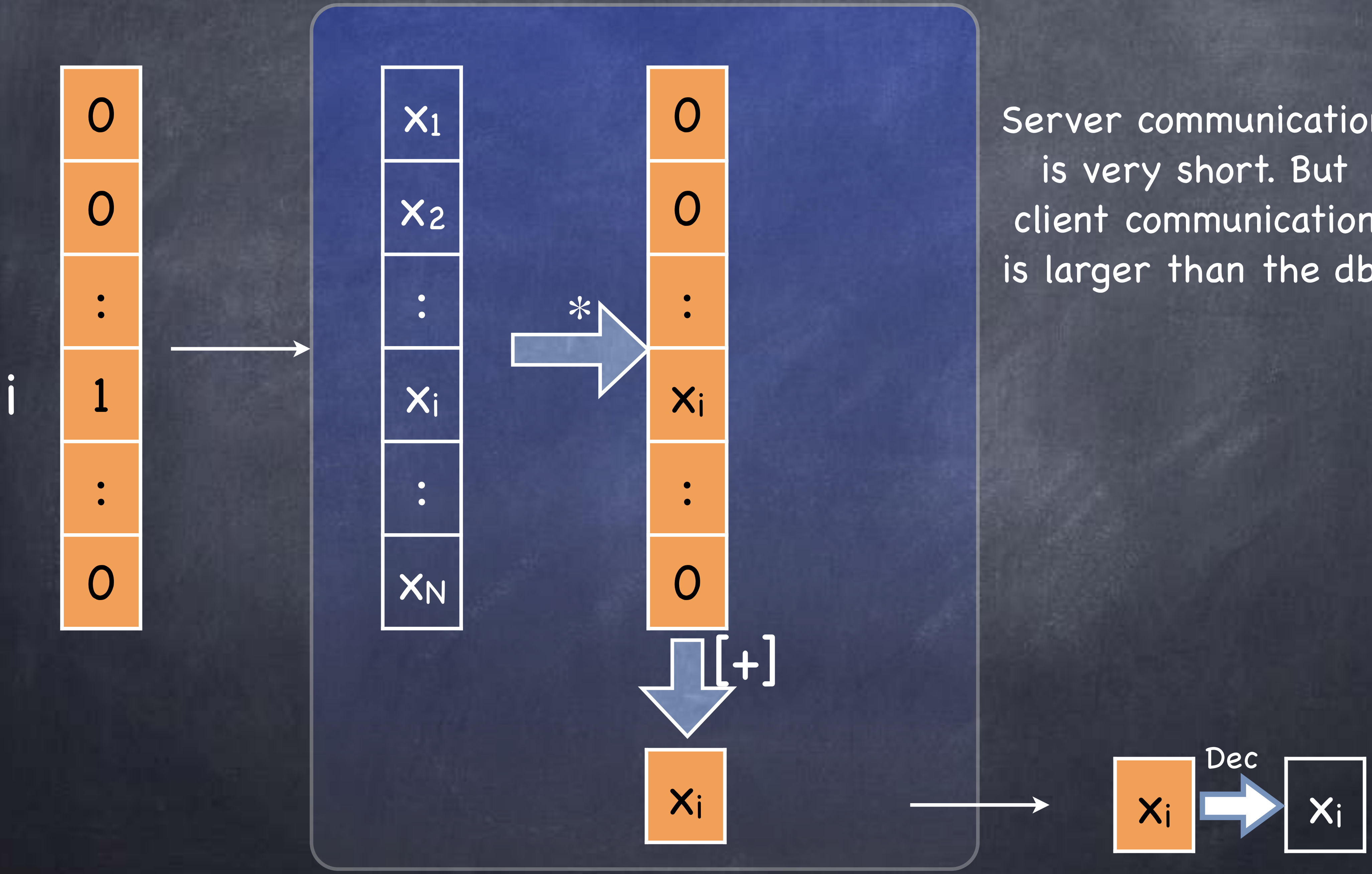
- In the following: database values are integers in [0,m), and we can use any homomorphic encryption scheme with a message space isomorphic with $\mathbb{Z}_n$ with $n \geq m$
  - e.g., Paillier encryption with message space $\mathbb{Z}_n$ ($n \geq m$)



- For integer a and ciphertext $\underline{c}$, can define $a*\underline{c}$ recursively: $0*\underline{c} = E(0)$; $1*\underline{c} = \underline{c}$; $(a+b)*\underline{c} = a*\underline{c}$ [+] $b*\underline{c}$.

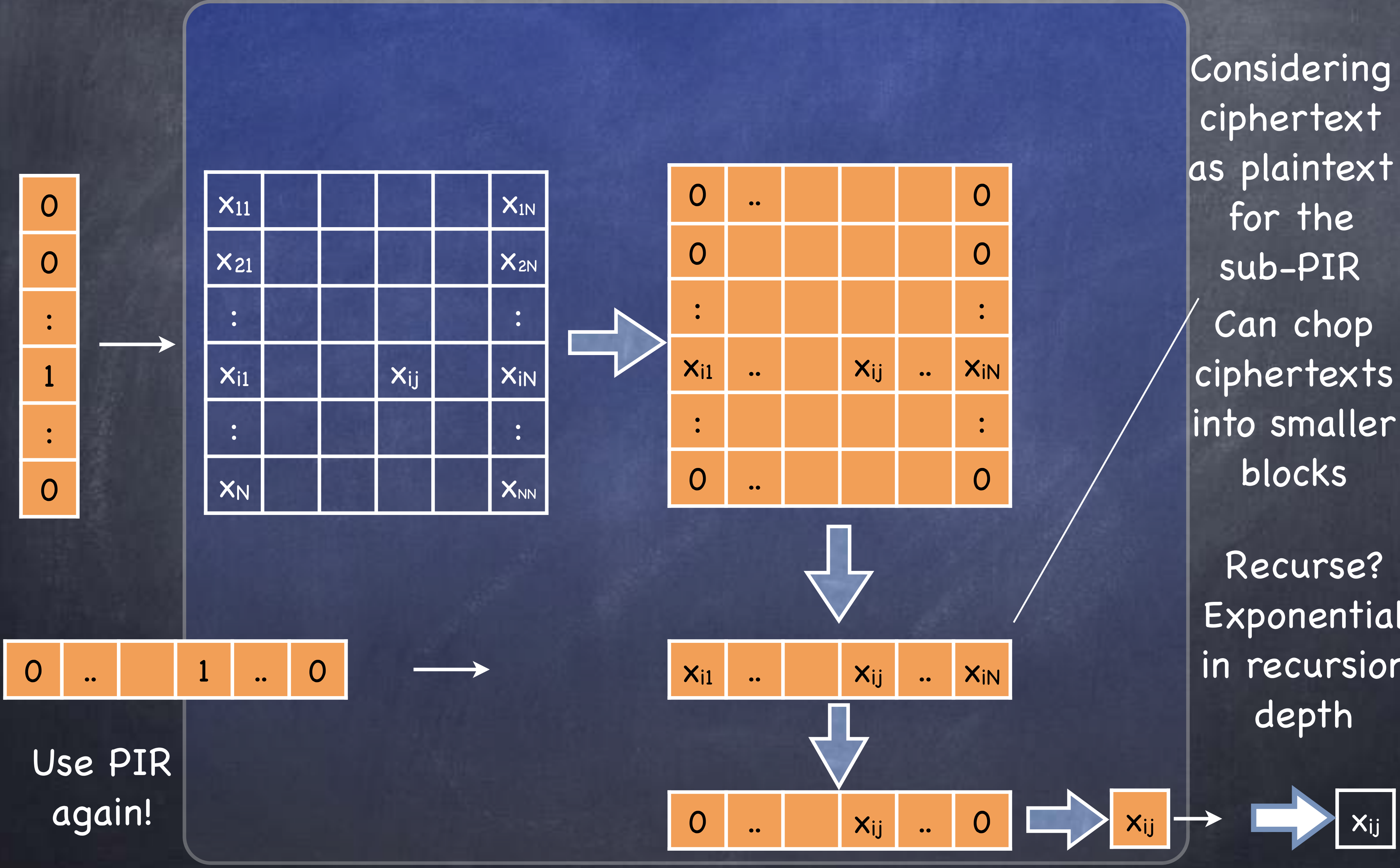


Private Information Retrieval



Server communication is very short. But client communication is larger than the db!

Private Information Retrieval



Considering ciphertext as plaintext for the sub-PIR

Can chop ciphertexts into smaller blocks

Recurse? Exponential in recursion depth

Use PIR again!

Private Information Retrieval

- Can dramatically improve efficiency if we have an efficient "recursive" homomorphic encryption scheme where:
  - Ciphertext in one level is plaintext in the next level
    - In Paillier, public-key (i.e., n) fixes the group for homomorphic operation (i.e., $\mathbb{Z}_n$)
  - Ciphertext size increases only "additively" from level to level
    - In Paillier, size of ciphertext about double that of the plaintext.
- Such a scheme: Damgård-Jurik cryptosystem

Damgård-Jurik Scheme

- Uses $\mathbb{Z}_{n(s+1)}^* \simeq \mathbb{Z}_{ns} \times \mathbb{Z}_n^*$, n=pq as in Paillier Encryption
  - Isomorphism: $\psi_s(a,b) = g^a b^{n^s}$ where g=(1+n)
- ψ_s can still be efficiently inverted if p,q known (but more involved)
- Recall Decisional Composite Residuosity assumption: Given n=pq (but not p,q), $\psi_1(0,\text{rand})$ looks like $\psi_1(\text{rand},\text{rand})$
- Enc(m) = $\psi_s(m,r)$ for m in $\mathbb{Z}_{ns}$ and a random r in $\mathbb{Z}_n^*$
- Homomorphism: Enc(m).Enc(m') is Enc(m+m')

  in $\mathbb{Z}_{n(s+1)}^*$

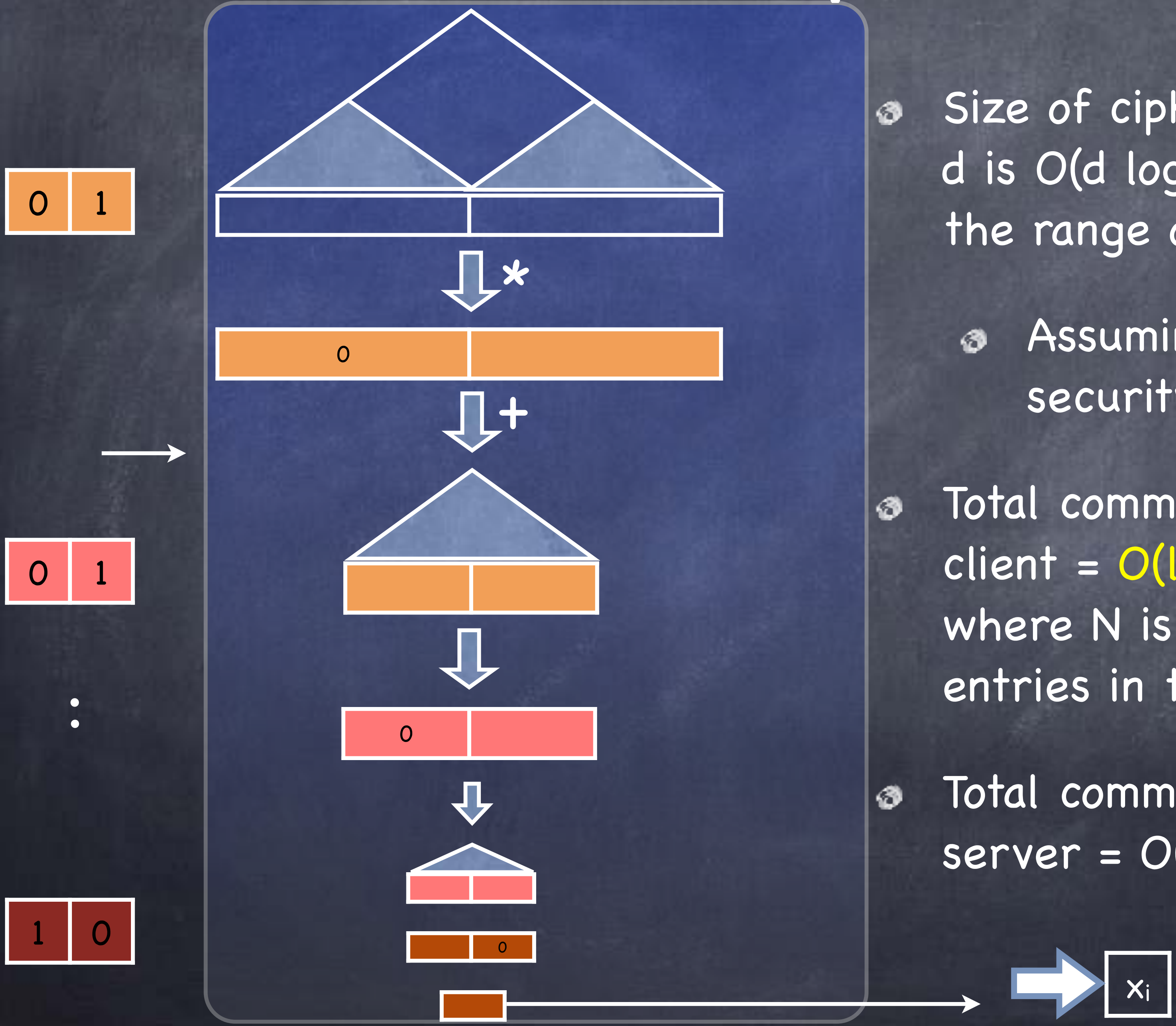
  - $\psi_s(m,r).\psi_s(m',r') = \psi_s(m+m',r.r')$

  in $\mathbb{Z}_{ns}$

- Recursive encryption: Output (ciphertext) of ψ_s ($\mathbb{Z}_{n(s+1)}^*$) is an input (plaintext) for ψ_{s+1} ($\mathbb{Z}_{n(s+1)}$) for the same public-key n.

  Note: s log n bits encrypted to (s+1)log n bits.
- IND-CPA secure under DCR (same as for Paillier)
- Unlinkability and multiplication by plaintext as in Paillier

Final PIR protocol



- Size of ciphertext at depth d is O(d log m) where m is the range of values in DB
  - Assuming log m ≥ security parameter
- Total communication from client = $O(\log^2 N \log m)$, where N is the number of entries in the DB
- Total communication from server = $O(\log N \log m)$