MTAT.07.006 Research Seminar in Cryptography

Single-Database Private Information Retrieval
07.11.2005

Aleksandr Grebennik
Tartu University
a_g@ut.ee

Overview of the Lecture

- CMS - first single database private information retrieval scheme
- Gentry-Ramzan PBR
- Lipmaa Oblivious Transfer Protocol with Log-Squared Communication

PIR, PBR

- PIR - allows a user to retrieve the i^{th} bit of an n-bit database, without revealing the value of index i to the database.

- PBR - natural and more practical extension of PIR in which, instead of retrieving only a single bit, the user retrieves a i^{th} block with d bits in it.

CMS - first single-database PIR

- Proposed by Cachin, Micali and Stadler in 1999

- Based on "Φ - hiding" assumption (that it is hard to distinguish which of two primes divide $\phi(m)$ for composite modulus m).

- Communication complexity is about $\mathcal{O}(\log^8 n)$ per bit.

CMS - first single-database PIR, slide 2

- Each index $j \in [1, n]$ is mapped to a distinct prime p_j.
- Query for bit b_i: hard-to-factor modulus m so that $p_i | \phi(m)$ and a generator $x \in \mathbb{Z}_m^*$.
- Server response: $r = x^P \mod m$, where $P = \prod_j p_j^{b_j}$
- Response retrieval: $\exists y : y^{p_i} \equiv r \pmod{m} \Leftrightarrow b_i = 1$

Gentry-Ramzan private block retrieval scheme

- Published in 2005

- Uses the fact that discrete logarithm computation is feasible in hidden subgroups of *smooth* order, while this task is still hard in general groups. (A number is called *smooth* if it has only *small* prime factors)

Gentry-Ramzan private block retrieval scheme, slide 2

- The server partitions the n-bit database B into t blocks $B = C_1 \| C_2 \| \ldots \| C_t$ of size at most ℓ bits.

- $S = \{p_1, \ldots, p_t\}$ is a set of small distinct prime numbers.

- Each block C_i is associated to a prime power π_i ($\pi_i = p_i^{c_i}$, where c_i is the smallest integer so that $p_i^{c_i} \geq 2^\ell$)

- All parameters above are public.

Gentry-Ramzan private block retrieval scheme, slide 3

- Server precomputes an integer e that satisfies $e \equiv C_i \pmod{\pi_i}$ using Chinese Remainder Theorem.

- To retrieve C_i it suffices to retrieve $e \mod \pi_i$.

Gentry-Ramzan private block retrieval scheme, slide 4

- To query for block C_i, the user generates an appropriate cyclic group $G = \langle g \rangle$ with order $|G| = q\pi_i$ for some suitable integer q and sends (G, g) to server, keeping q private.

- Example: an $\mathbb{Z}_m^*$ group, where m is constructed to Φ - hide π_i.

  ⋆ $m = Q_0 Q_1$, where Q_0, Q_1 are safe primes: $Q_0 = 2q_0\pi_i + 1, Q_1 = 2q_1 d + 1$; q_0, q_1 are primes.

- Notice that G contains a subgroup H of *smooth* order π_i, and that $h = g^q$ is a generator of H.

Gentry-Ramzan private block retrieval scheme, slide 5

- Server responds with $g_e = g^e \in G$

- The user obtains $e \mod \pi_i$ by setting $h_e = g_e^q \in H$ and performing a (tractable) discrete logarithm computation $\log_h h_e$, which occurs entirely in the subgroup H of order $p_i^{c_i}$ and can be quite efficient if p_i is small.

- To prove that $\log_h h_e = C_i$, let's rewrite $e \equiv e_{\pi_i} \pmod{\pi_i}$ as $e = e_{\pi_i} + \pi_i \cdot E$, for some $E \in \mathbb{Z}$. Now:

- $h_e = g_e^q = g_e^{|\langle g \rangle|/\pi_i} = g^{e|\langle g \rangle|/\pi_i} = g^{e_{\pi_i}|\langle g \rangle|/\pi_i} g^{E|\langle g \rangle|} = g^{e_{\pi_i}|\langle g \rangle|/\pi_i} = h^{e_{\pi_i}}$.

Gentry-Ramzan private block retrieval scheme, slide 6

- Pohlig-Hellman algorithm

- let's write $C_i = \log_h h_e$ in base p_i (remember that C_i is a number modulo $p_i^{c_i}$): $C_i = x_0 + x_1 p + \ldots x_{c-1} p^{c-1}, 0 \le x_i < p$

Gentry-Ramzan private block retrieval scheme, slide 7

- Computational complexity
  - $\star$ Querier side: no more than $4\sqrt{n\ell}$ group operations.
  - $\star$ Server side: $\Theta(n)$ group operations.
- Communication complexity
  - $\star$ Suppose that the group G and any element of G can be described in ℓ_G bits. Then the total complexity is $3\ell_G$ bits.

Lipmaa PIR protocol with log-squared communication

- first published in 2004

- Takes advantage of the concept of length-flexible additively homomorphic (LFAH) public-key cryptosystems.

  ⋆ *Length-flexible* public-key cryptosystem has an additional length parameter $s \in \mathbb{Z}^+$. The encryption algorithm maps sk-bit plaintexts, for any s and for security parameter k, to $(s+\xi)k$ -bit ciphertexts for some small integer $\xi \geq q$.

Lipmaa PIR protocol with log-squared communication

- Communication complexity
  - $\star$ $\Theta(k \log^2 n + \ell \log n)$
  - $\star$ $k = \Omega(log^{3-o(1)} n)$;
- Computational complexity
  - $\star$ Sender's work is equivalent to $\Theta(nl) \cdot k^{2+o(1)}$ bit operations;
  - $\star$ Receiver's work is $\Theta((k \cdot \log n + l)^{2+o(1)})$

Lipmaa PIR protocol with log-squared communication

- Communication complexity
  - ⋆ The ratio of amount of bits transferred to the communication complexity is $1/(\log n)$
  - ⋆ to achieve a good rate in practice, n and ℓ must be quite large (on the order of gigabits and megabits, respectively), before they begin to offset the large one-time cost represented by the $k \log^2 n$ term.
- Computational complexity
  - ⋆ Sender's work is equivalent to $\Theta(nl) \cdot k^{2+o(1)}$ bit operations;
  - ⋆ Receiver's work is $\Theta((k \cdot \log n + l)^{2+o(1)})$