Outsourced Pattern Matching

Carmit Hazay
Faculty of Engineering
Bar-Ilan University

Bar-Ilan University
Tradition of Excellence

Pattern Matching

- **Classic search problem:**
  - Given a text **T** and a pattern **P**, find all **exact** matched text locations

- In distributed systems text and pattern are given to distinct users

- Widely studied in the 70's and solvable in linear time[KMP77,BM77]
  - Many potential applications!

Distributed Pattern Matching in the Non-Private Setting

Receiver learns text positions I such that $\{p=T_i\}_{i\in I}$

T such that |T|=n

p such that |p|=m

p

matched positions

Sender

Receiver

Secure Pattern Matching

- In a secure variant sender does not learn anything about the pattern, while receiver does not learn anything about the other text locations

- Existing algorithms violate privacy when implemented distributively!

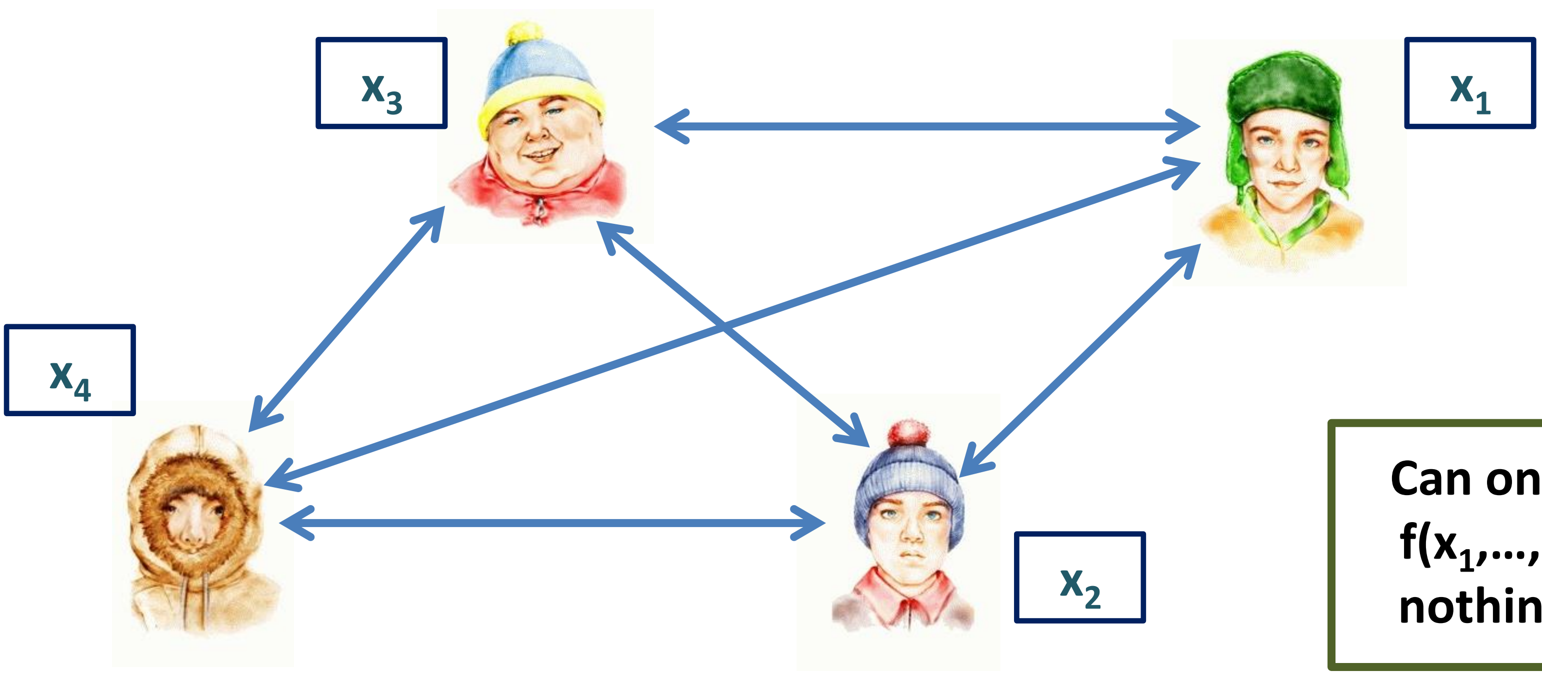
Secure Pattern Matching

- Many important applications: DNA matching

"You don't look anything like the long haired, skinny kid I married 25 years ago. I need a DNA sample to make sure it's still you."

Secure Computation

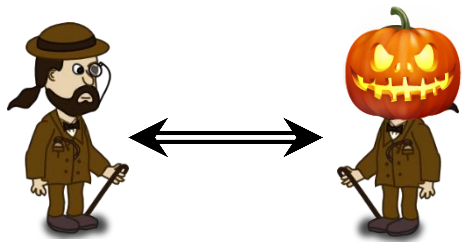
- A set of parties with inputs $x_1,\ldots,x_n$ that wish to determine $f(x_1,\ldots,x_n)$ for some function f



Can only learn $f(x_1,\ldots,x_n)$ and nothing more

Defining Secure 2P Computation

REAL

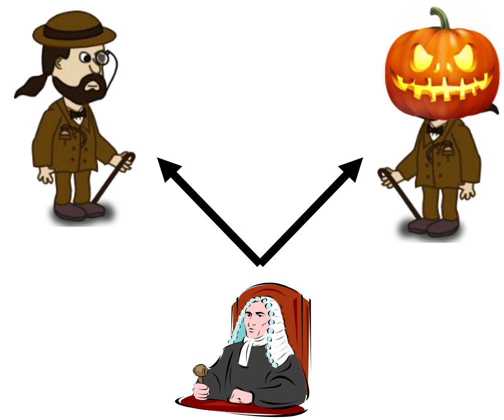


Protocol interaction

For every real adversary A

$\approx$

IDEAL



Trusted party

there exists an adversary S

Secure Computation

- This model is general enough to capture any cryptographic task such as:
  - Coin flipping
  - Electronic voting
  - Auctions with private bids

Secure Pattern Matching

- Known solutions: [TKC07, HL08, KM10, GHS10] use oblivious PRF, oblivious automaton evaluation and even garbling

- State-of-the-art protocol: [HT10] uses special type encoding
  - Overhead is linear in the text length

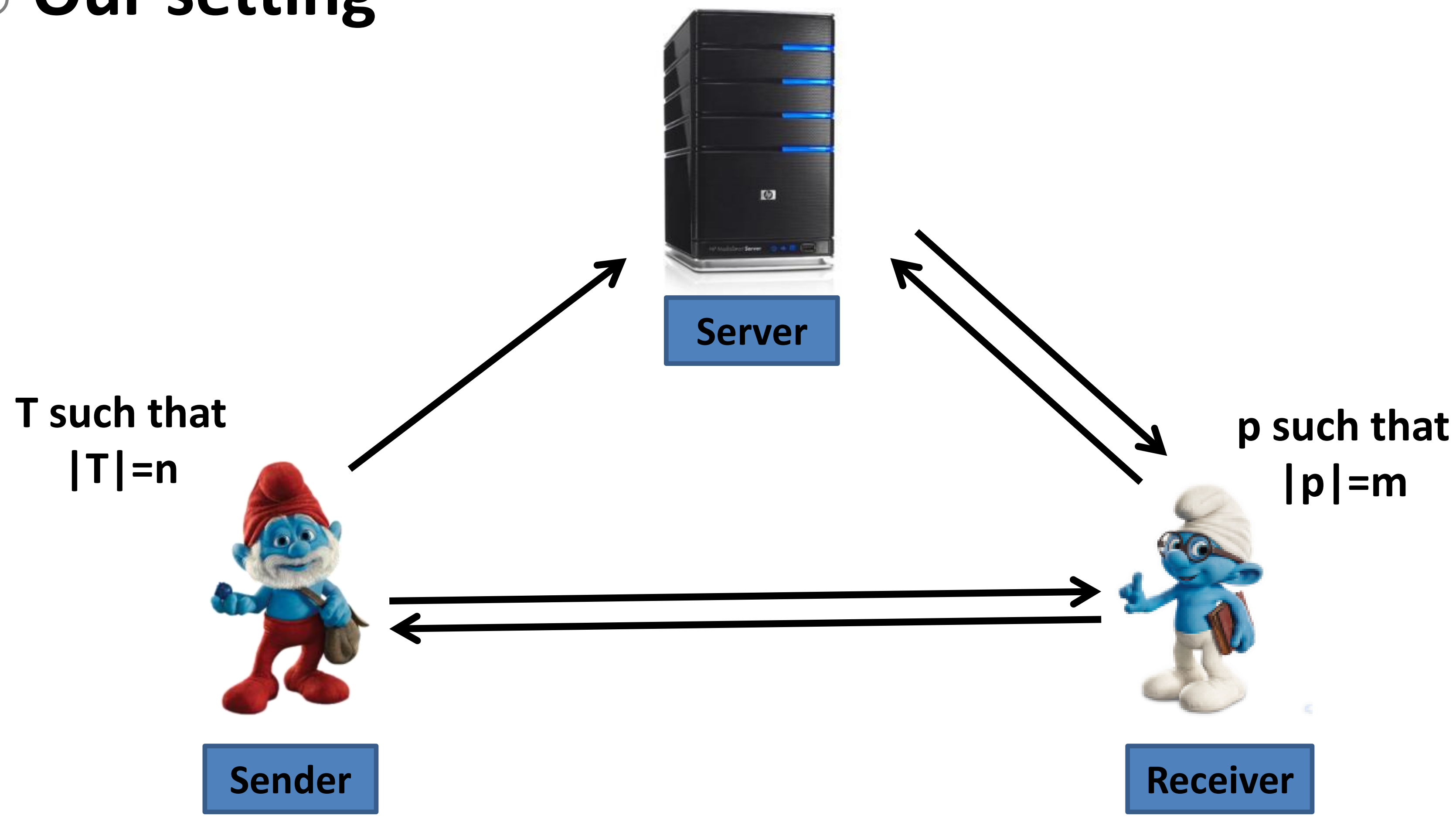
- What about other models?

Outsourced Secure Computation

- Resources are not evenly distributed
  - Powerful servers can provide storage and computation services
  - Cloud services are practically everywhere!

Outsourced Pattern Matching

- Our setting



Outsourced Pattern Matching

Two phases:

Preprocessing phase: sender sends **single** message to the server of preprocessed text

Query phase: receiver interacts with sender and then with the receiver

Efficiency:

Round optimal: minimal number of messages

Communication optimal: preprocessing costs **O(n)** bits query phase costs **O(m+|number of matches|)**

Outsourced Pattern Matching

In the query phase:

sender's state is **o(n)** (and even **O**(**k**))

Simulation-based security:

Server learns number of matches from the message size to the receiver

Server may collude with either receiver/sender

Outsourced Pattern Matching

- **Solution with small communication [FHV13]**
  - Non-standard assumption
- **Follow up work [HZ]**
  - Impossible under standard assumptions
- **New Result [H]**
  - Efficient solution with restricted collusion scenario

Outsourced Pattern Matching

[FaustHazayVenturi13]

Semi-Honest Outsourced Pattern Matching [FHV13]

- **Preprocessing:** Sender encodes positions of the substring $\mathbf{p_i}$ in the text by random values condition that they sum to **R**

- **Query phase:**

1) Sender hands the receiver trapdoor **R**

2) Server solves subset sum instance

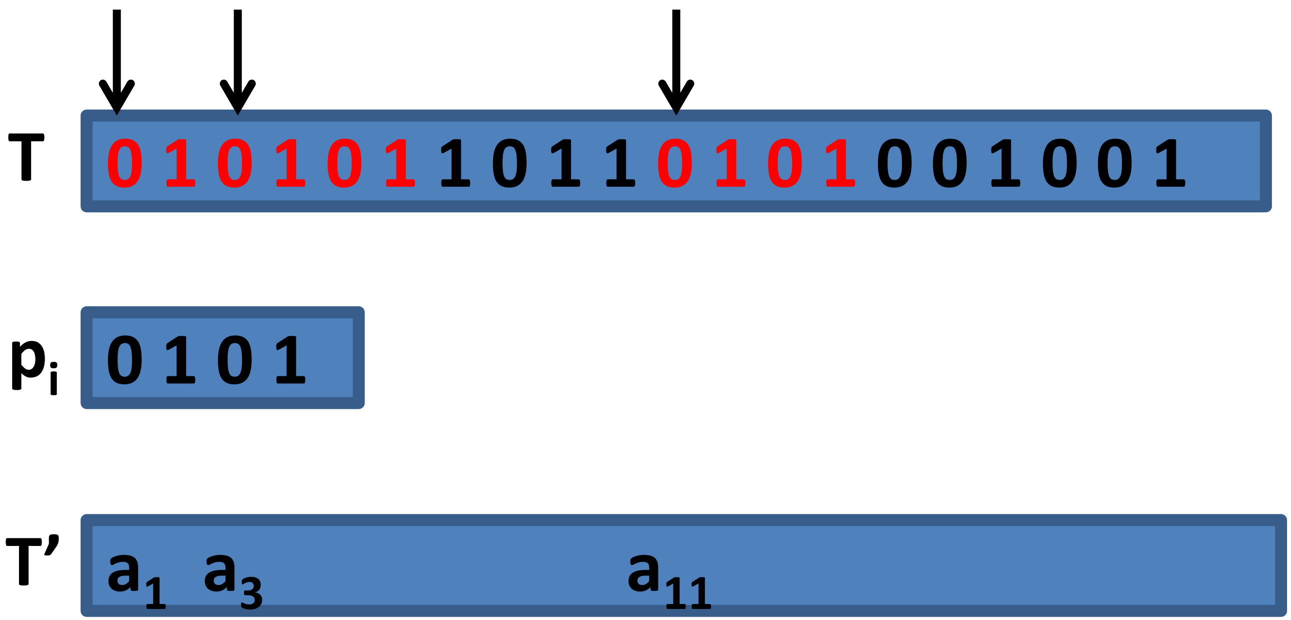
  - Requires easy instances of subset sum

Outsourced Pattern Matching [FHV13]

- The **subset sum problem** is parameterized with two integers **L** and **M**
- Random instance is defined by $(\mathbf{a}, \mathbf{R} = \mathbf{a}^{\top} \cdot \mathbf{s} \bmod \mathbf{M})$ for $\mathbf{a} \leftarrow \mathbf{Z}_{\mathbf{M}}^{\mathbf{L}}$ and $\mathbf{s} \in \{\mathbf{0},\mathbf{1}\}^{L}$

  Find **s** given **a** and **R**
- Hardness depends in **L/log M**
  - Easy when ratio smaller than **1/L** or greater than $\mathbf{L/log^2L}$

Outsourced Pattern Matching [FHV13]



Define a subset sum vector T' such that $a_1+a_3+a_{11}=F_k(0101)$

Outsourced Pattern Matching [FHV13]

- **Problem**: communication in query phase grows linearly with **n**
  - Otherwise subset sum parameters imply many collisions

- **Solution:** break the text into smaller subsets

Outsourced Pattern Matching [FHV13]

T 0 1 0 1 0 1 1 0 1 1 0 1 0 1 0 0 1 0 0 1

n = 20, m =4

(1) Break T into substrings of length 2m that overlap with m bits

b_1 0 1 0 1 0 1 1 0

b_2 0 1 1 0 1 1 0 1

b_3 1 1 0 1 0 1 0 0

b_4 0 1 0 0 1 0 0 1

Outsourced Pattern Matching [FHV13]

b_1 0 1 0 1 0 1 1 0

b_2 0 1 1 0 1 1 0 1

b_3 1 1 0 1 0 1 0 0

b_4 0 1 0 0 1 0 0 1

Requires a different trapdoor for each package!

(2) Pick a PRF key k and define a sequence of subset sum vectors

$a_1\ a_2\ a_3\ a_4\ a_5$ | $a_6\ a_7\ a_8\ a_9\ a_{10}$ | $a_{11}\ a_{12}\ a_{13}\ a_{14}\ a_{15}$ | $a_{16}\ a_{17}\ a_{18}\ a_{19}\ a_{20}$

such that:

$a_1+a_3 = F_k(0101||1)$, $a_2 = F_k(1010||1)$, $a_4 = F_k(1011||1)$, $a_5 = F_k(0110||1)$,
$a_{13} = F_k(0101||3)$

Outsourced Pattern Matching [FHV13]

b_1 0 1 0 1 0 1 1 0

b_2 0 1 1 0 1 1 0 1

b_3 1 1 0 1 0 1 0 0

b_4 0 1 0 0 1 0 0 1

Security proven in semi-honest model, malicious security is much more complicated

- Use a random oracle H to reduce trapdoor size
  - Program oracle's outcome and fix trapdoor to $\mathbf{F_k(p)}$

$\mathbf{a_1+a_3 = \mathit{H}(F_K(0101)\|1), a_2 = \mathit{H}(F_k(1010)\|1), a_4 = \mathit{H}(F_K(1011)\|1),}$
$\mathbf{a_5 = \mathit{H}(F_k(0110)\|1), a_6 = \mathit{H}(F_k(0110)\|2)}$

The Feasibility of Outsourced Database Search in the Plain Model [HazayZarosim]

The Feasibility of Outsourced Pattern Matching [HZ]

- **<u>Two results (semi-honest):</u>**
  1. **Impossibility of round and communication optimal protocols (applies to SSE as well)**
  2. **Abstraction of security properties of outsourced database search**

Infeasibility of Outsourced Pattern Matching [HZ]

- **Round and communication optimal cannot be achieved if:**
  1. Receiver colludes with the server
  2. Receiver sees preprocessed message from sender

- **Intuition:** simulator must commit to text before knowing receiver's queries
  - Needs to take into account too many options

Infeasibility of Outsourced Pattern Matching [HZ]

- Similar in spirit to infeasibility of non-interactive non-committing encryption [N02] but more complicated
  - When server and receiver collude need to ensure "right" order of interaction in query phase
  - Otherwise, communication depends on server's random tape

Infeasibility of Outsourced Pattern Matching [HZ]

- **Theorem:** either the receiver's random tape or message from sender is $\mathbf{O(n)}$

- **Cannot use PRGs to strengthen this result since reduction does not work**
  - Given a protocol $\boldsymbol{\pi}$ with long randomness $\mathbf{s}$ design a new protocol $\boldsymbol{\pi}'$ with randomness $\mathbf{G(r)}$
  - In the proof, reduce security of $\boldsymbol{\pi}'$ into security of $\boldsymbol{\pi}$ by invoking simulator of $\boldsymbol{\pi}$
  - Requires finding a preimage of $\mathbf{G}$!

Feasibility of Outsourced Pattern Matching [HZ]

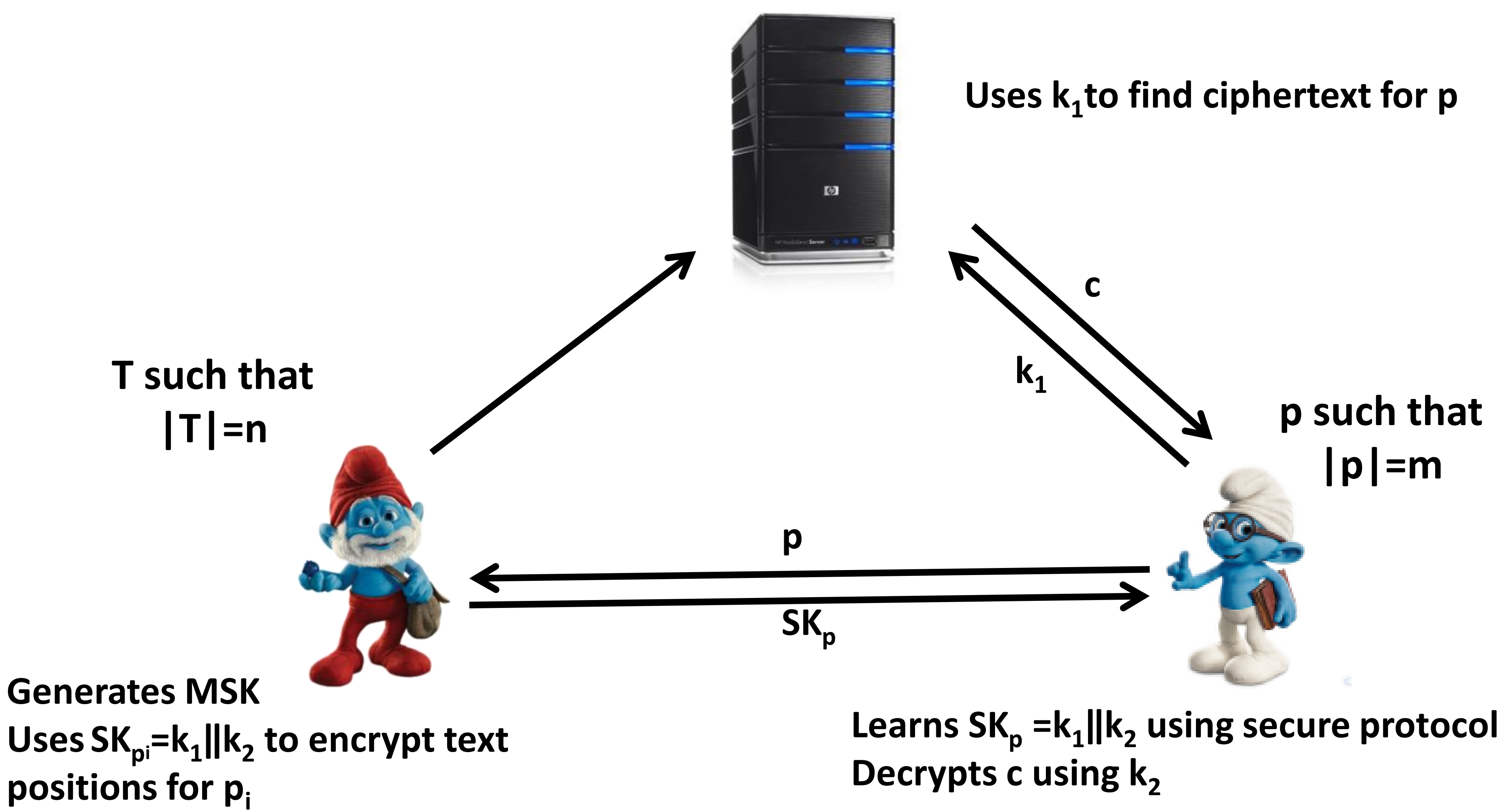
- Sender holds a master secret key **MSK** that generates subkey $\mathbf{SK_p}$ for each query **p** and uses it to encrypt matched text positions for **p**

- Sender holds both secret keys and database, thus can use symmetric key primitives like PRF

Feasibility of Outsourced Pattern Matching [HZ]

- Define encryption scheme with multiple subkeys, one per query with the properties:
  1. Semantically secure
  2. Secret key equivocation
     - Implies query privacy

Feasibility of Outsourced Pattern Matching [HZ]



Outsourced Pattern Matching – Revisited [Hazay]

Outsourced Pattern Matching – Revisited [H]

- Bypass the [HZ] impossibility result by restricting corruption scenarios
  - Server does not collude with the other parties

- **Advantages**:
  - Security in the presence of malicious server
  - Optimal communication and round complexity

Outsourced Pattern Matching – Revisited [H]

- Idea: use accumulators to ensure correctness
  - Sender stores all distinct elements of length **m** from **T** in an accumulator **Acc**
  - At most **n-m+1** elements

- For all $\mathbf{p_i} \in \mathbf{Acc}$ Sender encrypts all positions for which $\mathbf{p_i}$ matches the text
  - Prepands ciphertext **c** with **R** and appends it with $\mathbf{Mac_{k''}(c)}$ such that $\mathbf{R||k'||k''} = \mathbf{F_K(p_i)}$

Outsourced Pattern Matching – Revisited [H]

- In the query phase, the receiver learns $\mathbf{F_K(p)}$ using oblivious PRF evaluation protocol
- Let $\mathbf{R \| k' \| k''} = \mathbf{F_K(p)}$
  1. The receiver sends **R** to the server that finds ciphertext+tag that are prepended with **R**
  2. Server proves membership/non-membership relative to the accumulator
- The receiver verifies tag using **k''** and decrypts **c** using **k'**

Outsourced Pattern Matching – Revisited [H]

- **Security**:
  - Malicious server cannot claim that a string does not appear in **T** and cannot forge a tag
  - Semi-honest sender and receiver cannot learn additional information
    - Extension to malicious receiver is simple
    - Extension to malicious sender much harder

Future Research

- Better solutions with higher round complexity
- Extensions to related problems such as approximate pattern matching


Thank You!