Techniques for query verification

A brief overview

Matteo Loporchio

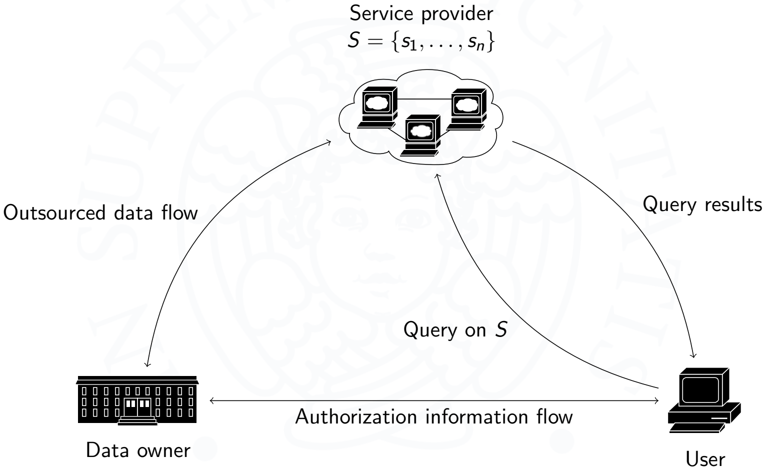
University of Pisa

January 22nd, 2021

Data outsourcing [LT18]

- The advent of cloud computing has opened new possibilities in terms of hardware and software resource distribution.
- Companies frequently delegate the storage and management of a data collection (e.g. a database) to a cloud provider.
- Providers have all the necessary infrastructures to make these data available to the public.
- This practice is commonly known as **data outsourcing**.
- Huge saving on maintenance costs, but we also face some security problems.

Data outsourcing [LT18]



Authenticated query processing

- Providers can return tampered or incomplete results.
- The **correctness** of the results is ensured if and only if:
  1. **Authenticity**. Results are returned without any modification.
  2. **Completeness**. All objects satisfying the query are returned.
- **Idea:** Force providers to send results + cryptographic proof (called **verification object**).
- Efficient retrieval, proof construction and verification.
- Proofs must be as succinct as possible, too!

Data model

- The collection $S = \{S_1, \ldots, S_m\}$ is a set of m sets.
- For $i \in \{1, \ldots, m\}$ it is $S_i \subseteq \mathbb{Z}_p$, with p prime.
- Given two indices $i, j \in \{1, \ldots, m\}$:
  1. **Subset**. Check whether $S_i \subseteq S_j$.
  2. **Empty intersection**. Check whether $S_i \cap S_j = \emptyset$.
- The provider must be able to generate **subset** and **disjointness** proofs for the client.

Example

- Let $p = 11$ and $S = \{X_1, X_2, X_3, X_4\}$.
- Suppose that:

$$X_1 = \{1, 3, 7, 9\} \qquad X_2 = \{1, 2, 8, 9, 10\}$$
$$X_3 = \{1, 2, 4, 5, 7, 9\} \qquad X_4 = \{2, 8, 9\}$$

- Input: $i = 4$, $j = 2$, i.e. is $X_4 \subseteq X_2$?
- Result: True + a proof that $X_4 \subseteq X_2$.

Background

A **group** is a set $\mathbb{G}$ paired with an operation $*$ such that:

1. **Closure**. For any $a, b \in G$ it is $a * b \in \mathbb{G}$.
2. **Associativity**. For any $a, b, c \in \mathbb{G}$ it is $(a * b) * c = a * (b * c)$.
3. **Identity**. There exists $1 \in \mathbb{G}$ s.t. $1 * a = a * 1 = a$ for any $a \in \mathbb{G}$.
4. **Inverse**. For any $a \in \mathbb{G}$, there exists a^{-1} s.t. $a * a^{-1} = a^{-1} * a = 1$.

Background

- The **order** of $\mathbb{G}$ is the number of its elements.
- If $*$ is multiplication, $\mathbb{G}$ is called **multiplicative**.
- A **generator** $g \in \mathbb{G}$ is an element such that

$$\forall x \in \mathbb{G} \quad \exists n \in \mathbb{N} \quad \text{s.t.} \quad x = g^n$$

- A **cyclic** group has at least one generator.
- If p is prime, then $\mathbb{Z}_p^* = \{1, \ldots, p-1\}$ is cyclic.

Example

n	1	2	3	4	5	6	7	8	9	10
$2^n \mod 11$	2	4	8	5	10	9	7	3	6	1
$3^n \mod 11$	3	9	5	4	1	3	9	5	4	1
$6^n \mod 11$	6	3	7	9	10	5	8	4	2	1

Table: 2 and 6 are generators for $\mathbb{Z}_{11}^*$, 3 is not.

Bilinear pairings

- Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of the same order p.
- Let g be a generator of $\mathbb{G}$.
- A (symmetric) **bilinear pairing** is a function $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ such that:
  1. **Bilinearity**. For any $u, v \in \mathbb{G}$ and any $a, b \in \mathbb{Z}$ it is $e(u^a, v^b) = e(u, v)^{ab}$.
  2. **Non degeneracy.** $e(g, g) \neq 1$.
  3. **Computability.** For any $u, v \in \mathbb{G}$, computing $e(u, v)$ is efficient.

Elliptic curves

- The group $\mathbb{G}$ is typically an **elliptic curve**.
- Defined as a "cloud" of points in $\mathbb{Z}_p^2$. For $p > 3$:

$$\mathcal{E}_p(a,b) = \{(x,y) \in \mathbb{Z}_p^2 \mid y^2 \equiv x^3 + ax + b \mod p\} \cup \{O\}$$

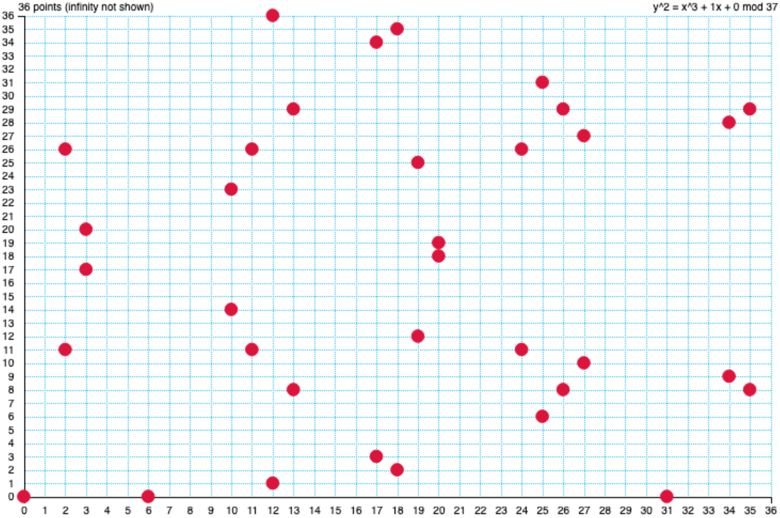
- The group operation in $\mathcal{E}_p(a,b)$ is **addition** between points.
- Not the usual "component-wise" addition!

$$(x_u, y_u) + (x_v, y_v) \neq (x_u + x_v, y_u + y_v)$$

- With **multiplicative notation**, for any $u \in \mathbb{G}$ and $n \in \mathbb{N}$:

$$u^n \quad \text{is equivalent to} \quad \underbrace{u + \ldots + u}_{n \text{ times}}$$

Elliptic curves



Set accumulators

- Let p be a prime number.
- Let $\mathbb{G}$ a cyclic group of order p and $g \in \mathbb{G}$ a generator.
- **Accumulator**: a function *acc* that maps a set to an element of $\mathbb{G}$.

$$acc : \mathcal{P}(\mathbb{Z}_p) \rightarrow \mathbb{G}$$

- Given a set X, we call $acc(X)$ the **accumulative value**.
- **Collision-resistance**: given a set X, it is difficult to find $Y \neq X$ such that $acc(Y) = acc(X)$.

Characteristic polynomial

- Given a set $X \subseteq \mathbb{Z}_p$, we call

$$P_X(z) = \prod_{x \in X} (x + z)$$

the **characteristic polynomial** of X, with coefficients in $\mathbb{Z}_p$.

- For any X it is $\deg P_X(z) = |X|$.
- Given $X = \{x_1, \ldots, x_n\}$, the coefficients $a_1, \ldots, a_n$ of

$$P_X(z) = \prod_{i=1}^{n} (x_i + z) = \sum_{i=0}^{n} a_i z^i$$

can be computed in $O(n \log n)$ time, as proved in [PS77].

Bilinear accumulator

- **Bilinear accumulator**:

$$acc(X) = g^{P_X(s)}$$

  where $s \in \mathbb{Z}_p$ is a random **secret** (trapdoor).
- Collision-resistant, as proved in [PTT15].
- We can still compute $acc(X)$ without knowing s if $(g, g^s, g^{s^2}, \ldots, g^{s^q})$ is public, with $q \geq |X|$.

Example

- Let $p = 11$, $g = 2$, $s = 3$ and $X = \{4, 7\} \subseteq \mathbb{Z}_{11}$.
- $P_X(z) = (z + 4)(z + 7) = z^2 + 6$.
- The accumulative value is:

$$acc(X) \equiv g^{P_X(s)} \equiv 2^{3^2+6} \equiv 2^{15} \equiv 10 \mod 11$$

- Suppose that $(g, g^s, g^{s^2}) \equiv (2, 8, 6) \mod 11$ is public.
- We can still compute $acc(X)$ as:

$$acc(X) \equiv g^{s^2+6} \equiv g^{s^2} \cdot g^6 \equiv 6 \cdot 2^6 \equiv 6 \cdot 9 \equiv 10 \mod 11$$

Security

- The security of accumulators hinges on the difficulty of solving a specific cryptographic problem.
- **Discrete logarithm**: if we know g and $k = g^s$ it is difficult to obtain $s = \log_g k$.
- q-**Strong Bilinear Diffie-Hellman problem**: given a tuple $(g, g^s, g^{s^2}, \dots, g^{s^q})$ as input, output the pair

$$(e(g,g)^{1/(s+x)}, x)$$

with $x \in \mathbb{Z}_p$.

Security

- *q*-**SBDH assumption**: for any PPT (probabilistic polynomial-time) algorithm $\mathcal{A}$ it is:

$$\mathcal{P}[\mathcal{A}(g, g^s, g^{s^2}, \ldots, g^{s^q}) = (e(g,g)^{1/(s+x)}, x)] \leq \epsilon$$

  where ϵ is a negligible quantity.
- As long as s is secret, a PPT algorithm cannot derive the output pair of the q-SBDH problem (except with a negligible probability).

Authentication protocols

- Our goal is to design **protocols** to authenticate subset and intersection operations.
- Protocols define the interaction between provider and client.
- The group $\mathbb{G}$, the pairing e and the prime p are public.
- The owner chooses $s \in \mathbb{Z}_p$ at random and keeps it **secret**.
- The owner **publishes** $pk = (g, g^s, g^{s^2}, \ldots, g^{s^q})$.

Subset operation [PTT11]

Let X and Y be two sets.

1. The provider computes $\pi = acc(Y \setminus X)$.
2. The provider sends π, $acc(X)$, and $acc(Y)$ to the client.
3. The client checks if

$$e(acc(X), \pi) \stackrel{?}{=} e(acc(Y), g)$$

4. If this is the case, then the proof is *valid* and the client can be sure that $X \subseteq Y$.

Subset operation [PTT11]

- Why is this protocol correct?
- Recall that $acc(X) = g^{P_X(s)}$ and $acc(Y) = g^{P_Y(s)}$.
- Suppose that $\pi = acc(Y \setminus X) = g^{P_{Y \setminus X}(s)}$.
- Then it is:

$$e(acc(X), \pi) = e(g^{P_X(s)}, g^{P_{Y \setminus X}(s)}) = e(g, g)^{P_X(s) \cdot P_{Y \setminus X}(s)}$$

- And also:

$$e(acc(Y), g) = e(g^{P_Y(s)}, g) = e(g, g)^{P_Y(s)}$$

- Then $e(g, g)^{P_X(s) \cdot P_{Y \setminus X}(s)} = e(g, g)^{P_Y(s)}$ if and only if:

$$P_X(s) \cdot P_{Y \setminus X}(s) = P_Y(s)$$

Example

- Let $p = 11$ and $pk \equiv (g, g^s, g^{s^2}, g^{s^3}) \equiv (2, 8, 6, 7) \mod 11$.
- Let $X = \{4\}$ and $Y = \{1, 4, 5\}$.
- We have that:

$$\begin{aligned} acc(X) &\equiv g^{P_X(s)} \equiv g^{s+4} && \equiv 7 \mod 11 \\ acc(Y) &\equiv g^{P_Y(s)} \equiv g^{s^3+10s^2+7s+9} && \equiv 7 \mod 11 \end{aligned}$$

- The provider computes $Y \setminus X = \{1, 5\}$ and $P_{Y \setminus X}(z) = z^2 + 6z + 5$.

Example

- Then it sends

$$\pi = acc(Y \setminus X) \equiv g^{s^2+6s+5} \equiv 6 \cdot 8^6 \cdot 2^5 \equiv 4 \mod 11$$

  and the values $acc(X) = acc(Y) = 7$ to the client.
- The client checks if the following equality holds:

$$e(acc(X), \pi) = e(7, 4) \stackrel{?}{=} e(7, 2) = e(acc(Y), g)$$

Empty intersection

Let X and Y be two sets. We can prove the following results:

- If $X \cap Y = \emptyset$ then $GCD(P_X, P_Y) = 1$.
- $P_{X \cap Y}(z) = GCD(P_X, P_Y)$.
- If $X \cap Y = \emptyset$ then there exist $U(z)$ and $V(z)$ such that:

$$U(z)P_X(z) + V(z)P_Y(z) = 1$$

- $U(z)$ and $V(z)$ can always be obtained by means of the **Extended Euclidean Algorithm**.

Empty intersection

Let X and Y be two disjoint sets.

1. The provider computes $\pi_1 = g^{U(s)}$ and $\pi_2 = g^{V(s)}$ by exploiting the fact that:

$$U(s) \cdot P_X(s) + V(s) \cdot P_Y(s) = 1$$

2. The provider sends π_1, π_2, $acc(X)$ and $acc(Y)$ to the client.
3. The client checks if this condition holds:

$$e(\pi_1, acc(X)) \cdot e(\pi_2, acc(Y)) \stackrel{?}{=} e(g, g)$$

4. If this is the case, then the proof is *valid* and the client can be sure that $X \cap Y = \emptyset$.

Example

- Suppose that $p = 11$, $pk \equiv (g, g^s, g^{s^2}, g^{s^3}) \equiv (2, 8, 6, 7)$ mod 11, $X = \{1, 3, 4\}$, $Y = \{2, 5, 7\}$.
- The provider computes $P_X(z) = z^3 + 8z^2 + 8z + 1$ and $P_Y(z) = z^3 + 3z^2 + 4z + 4$ and the accumulative values:

$$acc(X) \equiv g^{P_X(s)} \equiv g^{s^3+8s^2+8s+1} \equiv 5 \mod 11$$
$$acc(Y) \equiv g^{P_Y(s)} \equiv g^{s^3+3s^2+4s+4} \equiv 1 \mod 11$$

- Given that X and Y are disjoint, it is:

$$\underbrace{(9z^2 + z + 4)}_{U(z)} \cdot P_X(z) + \underbrace{(2z^2 + 9z + 2)}_{V(z)} \cdot P_Y(z) = 1$$

Example

- The provider then computes:

$$\begin{aligned} \pi_1 &\equiv g^{U(s)} \equiv g^{9s^2+s+4} \equiv (g^{s^2})^9 \cdot g^s \cdot g^4 &&\equiv 3 \mod 11 \\ \pi_2 &\equiv g^{V(s)} \equiv g^{2s^2+9s+2} \equiv (g^{s^2})^2 \cdot (g^s)^9 \cdot g^2 &&\equiv 7 \mod 11 \end{aligned}$$

  and sends π_1, π_2, $acc(X)$, $acc(Y)$ to the client.
- The client verifies the proofs by checking if:

$$e(\pi_1, acc(X)) \cdot e(\pi_2, acc(Y)) = e(3,5) \cdot e(7,1) \stackrel{?}{=} e(2,2)$$

- Computing $e(u,v)$ is efficient for any $u, v \in \mathbb{G}$, so the verification phase is easy.

Unforgeability

- Set accumulators are considered secure if they are **unforgeable**.
- This means that a polynomial-time adversary $\mathcal{A}$ has a negligible success probability of forging fake proofs.
- Consider the following experiment:
  1. Give $pk = (g, g^s, g^{s^2} \ldots, g^{s^q})$ to $\mathcal{A}$.
  2. $\mathcal{A}$ outputs two sets X, Y with a disjointness proof (π_1, π_2).
- We say that $\mathcal{A}$ succeeds if and only if $X \cap Y \neq \emptyset$ and the client declares (π_1, π_2) as valid.
- The bilinear accumulator is unforgeable, as proved in [PTT11].

Conclusions

- Set accumulators can be used to generate constant-size proofs for queries on outsourced data collections.
- More “expressive” than Merkle Hash Trees (allow subset and disjointness proofs).
- Different protocols and accumulators for other set operations and more complex queries also exist.
- Recently used in the context of blockchain systems. [XZX19]
- Some important downsides:
  - Computational overhead (operations on elliptic curves).
  - Public key size is $O(q)$, with $q = \max_{S_i \in S} |S_i|$.

The end

Thank you for your attention.

References I

[LT18] Ling Liu, M. Tamer Özsu, "Encyclopedia of Database Systems, Second Edition" Springer-Verlag, New York, 2018.

[PS77] Franco P. Preparata, Dilip V. Sarwate. "Computational complexity of Fourier transforms over finite fields", Mathematics of Computation, vol. 31, n. 139, pp. 740-751, 1977.

[PTT11] Charalampos Papamanthou, Roberto Tamassia, Nikos Triandopoulos, "Optimal Verification of Operations on Dynamic Sets", Advances in Cryptology – CRYPTO 2011, Lecture Notes in Computer Science, vol. 6841, Springer, Berlin, Heidelberg, 2011.

[PTT15] Charalampos Papamanthou, Roberto Tamassia, Nikos Triandopoulos, "Authenticated hash tables based on cryptographic accumulators", Algorithmica, vol. 74, n. 2, pp. 664-712, 2015.

References II

[Tam03] Roberto Tamassia, "Authenticated Data Structures", Algorithms - ESA 2003, Lecture Notes in Computer Science, vol. 2832, Springer, Berlin, Heidelberg, 2003.

[XZX19] Cheng Xu, Ce Zhang, Jianliang Xu, "vChain: Enabling Verifiable Boolean Range Queries over Blockchain Databases", ACM SIGMOD, 2019.