Privacy and Integrity Preserving Computations with CRISP

Sylvain Chatel Apostolos Pyrgelis Juan Ramon Troncoso-Pastoriza Jean-Pierre Hubaux

École Polytechnique Fédérale de Lausanne

Overview

In recent years, users have received more and more incentives to share their data with **service providers** (SP).

But the sensitivity of personal data can lead to **privacy** concerns.

Fully Homomorphic Encryption (FHE) has recently emerged as an auspicious solution for privacy-preserving cloud analytics and outsourcing but **correctness** of the process is never guaranteed.

We propose a first building block to remediate this observation. **CRISP** enables authenticity verification of data encrypted under trending FHE schemes.

Our solution relies on a tailored circuit and the combination of **MPC-based proof systems**, **lattice-based commitments**, and **FHE**.

We implemented CRISP and evaluated its performance in different use-cases and identified several **trade-offs** to achieve **practicality**.

Motivation

As data are increasingly generated about users, incentives to share it with various service providers to obtain access to services and application has heightened.

For example, a user can obtain her sequenced genome from a medical centre and subsequently share it with a direct-to-consumer service interested in the user's susceptibility to specific diseases.

As genomic data represent critical sensitive information, the user can use homomorphic encryption to protect her data. But this protection hinders the service provider's ability to ensure the user is not cheating.

To solve this issue, we introduce CRISP to enable oblivious verification that encrypted data matches data issued by a trusted service provider.

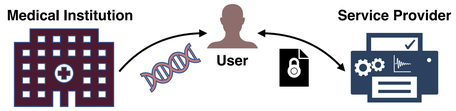

Figure 1. Genomic data analysis.

CRISP's workflow

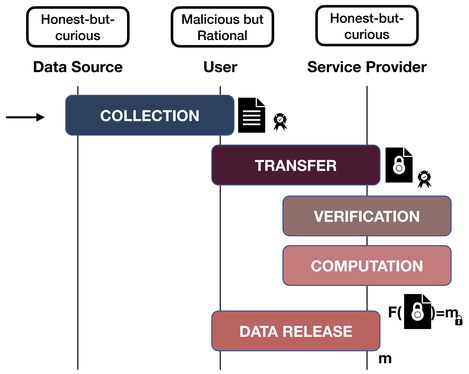

Figure 2. CRISP's interactions.

CRISP

CRISP relies on a tailored combination of FHE [4], lattice-based commitments (BDLOP [1]), and Multiparty Computation in-the-head (MPCitH [5, 2]) to ensure security and privacy.

Privacy

Users are guaranteed confidentiality of their data:
→ Semantic security of FHE scheme.
→ Hiding of BDLOP.
→ Zero-knowledge proof.
→ Security of keyed hash functions.

Security

Service providers are convinced the encrypted data is correct:
→ Correctness of the FHE scheme.
→ Binding of the BDLOP.
→ Soundness MPC in-the-head.
→ Security of existing hash-based signature schemes.

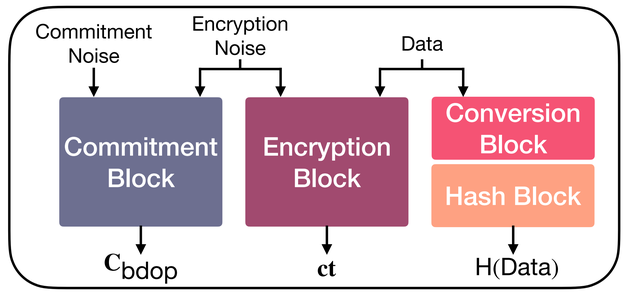

Figure 3. CRISP's tailored arithmetic circuit.

CRISP's Transfer and Verification Phases

Proof of correct encryption can be achieved using a specific kind of proof system names **Multiparty Computation in-the-Head**. By emulating several (virtual) parties, a prover can run MPC protocol equivalent to the circuit and it also commits to the state of each of the virtual parties. The verifier challenges a subset of those parties to be opened and check their consistency.

The advantage of this proof system is its prover efficiency and its native algebra being compatible with the lattice-based constructions: i.e., **encryption** and **BDLOP commitment schemes**.

Challenges

1. **Compatibility between the arithmetic rings:**
   Current efficient FHE schemes, BDLOP, and the hash function are represented each on different finite fields.
   ▷ Design of custom conversion blocks used in our tailored circuit Fig 3.
2. **Proof size:**
   MPC in-the-head induced a proof size linear in the input of the circuit.
   ▷ Several optimizations to reduce the proof size.
3. **Compatibility with existing deployments of crypto suites:**
   Recent very efficient and arithmetically friendly hash functions are not yet widely available.
   ▷ Use of SHA-256 as the keyed hash function.

Practical Evaluation

We implemented and evaluated CRISP on different use-cases.

- **A: Genomic data analysis**
  Weighted sum with few data points.
- **B: Smart metering**
  Large amount of data aggregation.
- **C: Personal Activity Tracking**
  Polynomial approximation of a non-linear function.

Table 1. CRISP's Evaluation.

Case	$\log N$	Size (MB)	t_{prove} (s)	t_{ver} (s)
A	11	12	8.2	4.2
B	12	203	63	26
C	13	472	149	104

Trade-offs

Few optimizations:

- **RIC**:
  Random Checks
- **BG**:
  Batching
- **PP**:
  Pre-processing

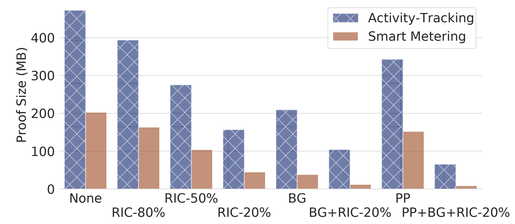

Figure 4. Proof size optimization.

Conclusion

CRISP is a generic solution that protects the interests of both users and service providers. Building on state-of-the-art lattice-based homomorphic encryption and commitments, as well as zero-knowledge proofs, it enables users to offload their data to service providers in a privacy and integrity preserving manner, yet still enables flexible computations on it.

Paper:

Current Projects:

References

[1] C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner, and C. Peikert.
More Efficient Commitments from Structured Lattice Assumptions.
In *SCN*, 2018.

[2] M. Chase, D. Derler, S. Goldfeder, C. Orlandi, S. Ramacher, C. Rechberger, D. Slamanig, and G. Zaverucha.
Post-Quantum Zero-Knowledge and Signatures from Symmetric-Key Primitives.
In *ACM SIGSAC CCS*, 2017.

[3] S. Chatel, A. Pyrgelis, J. R. Troncoso-Pastoriza, and J.-P. Hubaux.
Privacy and integrity preserving computations with CRISP.
In *USENIX Security*, 2021.

[4] J. H. Cheon, A. Kim, M. Kim, and Y. Song.
Homomorphic encryption for arithmetic of approximate numbers.
In *ASIACRYPT*, 2017.

[5] I. Giacomelli, J. Madsen, and C. Orlandi.
ZKBoo: Faster zero-knowledge for boolean circuits.
In *USENIX Security Symposium*, 2016.