Compare Before You Buy

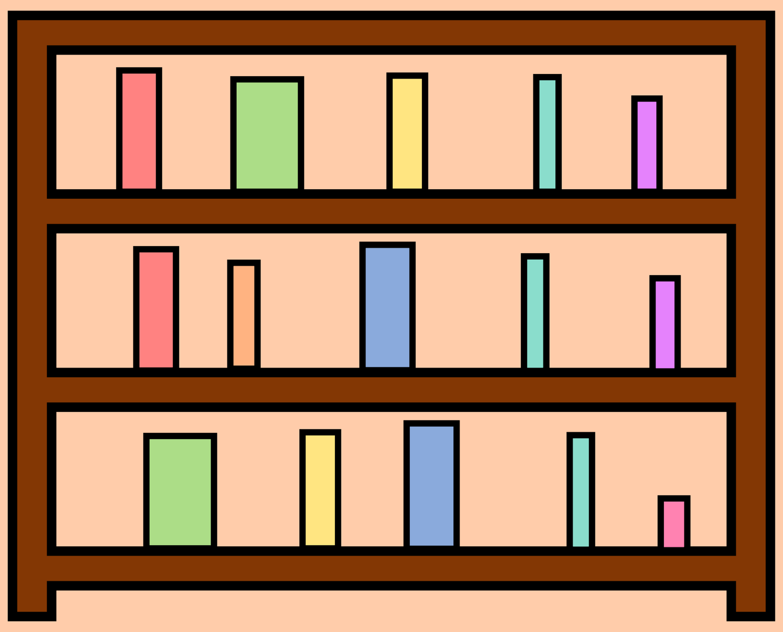
Privacy-preserving selection of threat intelligence providers

Jelle Vos, Christian Doerr, Zekeriya Erkin

INTRODUCTION

Cyber criminals often reuse the same resources for their attacks in order to press costs, so we can collect special indicators that reveal criminal activity. Organizations who want to protect themselves against cyber crime buy these indicators in the form of cyber threat intelligence (CTI) from CTI providers. Ideally, **an organization buys intelligence from multiple providers**, for example to prevent biased data. *However, how does a company prevent buying the same indicators twice?*

Our work presents a **privacy-preserving protocol** for approximating the **number of distinct elements** among multiple sets. Specifically, we provide a solution to the multi-party private set union-cardinality (MPSU-CA) problem. Our protocol requires less communication rounds than previous protocols, so CTI providers can participate with minimal effort.



Shopping for threat intelligence is like shopping for books, but you are only allowed to buy entire shelves. Even worse: You do not know which books you are buying. In this case, if you are buying all three shelves, you are getting 15 books, but only 8 unique ones. *Are you getting your money's worth?*

METHOD

ElGamal is a homomorphic cryptosystem that lets perform arithmetic operations on ciphertexts. As the underlying group, we choose **elliptic curves**, which are several orders of magnitude faster than integer groups for the same level of security. We propose a **sub-protocol that securely shuffles and decrypts** these ciphertexts, so that no party knows the original ordering of the plaintexts.

Since there are 2^{32} IP addresses, it is infeasible to count each possible IP address separately. Instead, we let each party encode their set as a Bloom filter:

```
xxx.xxx.xxx.xxx
| 0 | 0 | 0 | 0 | R | 0 | 0 | R | 0 | 0 |
```

The Bloom filter is filled with 0s. Each IP address sets h bins to a random value. After the whole set is encoded, the party encrypts the Bloom filter.

We then compute the union by summing the Bloom filters homomorphically.

From the resulting Bloom filter we can **estimate** how many **unique elements** were put into it by counting how many bins are set to a random value. Before we do, the parties **shuffle** the ciphertexts **so the positions of the random values do not reveal information** about the actual addresses. Instead of including every IP address, we can also ignore p% of the addresses to lower the size of the filter.

RESULTS

We implemented our protocol in **Rust**. We evaluate the run time of the protocol for different parameters. Consider five parties with 20,000 IP addresses, and 50,000 unique IP addresses, then the table on the right captures the measured run times and estimates over 20 experiments.

	Lower accuracy			Higher accuracy		
	p = 100%	p = 50%	p = 25%	p = 100%	p = 50%	p = 25%
Bins m	10,000	5,000	2,500	50,000	25,000	12,500
Encryption [s]	1.5	0.8	0.4	17.5	8.7	4.4
Shuffle-decrypt [s]	12.0	6.1	3.0	60.0	30.4	15.0
Estimate mean	49,539	49,510	50,771	50,066	50,081	50,065
Standard deviation	±1,284	±1,364	±2,381	±165	±307	±490

Results for five parties with 20,000 IP addresses each. Combined, they have 50,000 unique IP addresses. We see that the accuracy of the estimate decreases when p is smaller, and so does the run time. The same holds when the number of bins m decreases. In conclusion, there is a trade-off between run time and accuracy.

CONCLUSION

We summarize our contributions as follows:

- A provably secure, efficient MPSU-CA protocol
- The protocol requires fewer interactions than previous protocols, since parties only have to encrypt the set once
- An open-source proof-of-concept implementation

TU Delft