Privacy-friendly machine learning algorithms for intrusion detection systems

Supervisor: Pr. dr. ir. Bart Preneel

Henri De Plaen

Master Applied Mathematics, KU Leuven

December 22, 2017 Leuven, Belgium

1. Introduction
2. Intrusion detection systems
3. Multiparty computation
4. Methodology
5. Conclusion

- As computers are used more and more (including for sensible data), and the connection between them increases as well, there is a constant need for better intrusion detection systems.
- Machine learning algorithms can increase performance of a lot of existing applications, but they need a significant dataset.
- The amount of data increases as well, but remains sensible in the case of intrusion detection systems, hence the need for encryption.

Privacy-friendly data pooling for enhancing intrusion detection systems

Different types, based on where the intrusion takes place

- Network Intrusion Detection System (NIDS)
- Host Intrusion Detection System (HIDS)
- Hybrid Intrusion Detection System

Different detection methods

- Signature based
  - Advantages: accuracy and time
  - Disadvantages: only known intrusion types are detected
- Anomaly based
  - Advantages: new intrusion types can be detected
  - Disadvantages: malicious activity disguised as normal traffic can pass through
- Machine learning (classification)

Different types, based on where the intrusion takes place

- Network Intrusion Detection System (NIDS)
  - Advantages: detects attack before it occurs
  - Disadvantages: needs to be implemented on the network
- Host Intrusion Detection System (HIDS)
  - Advantages: collects broader data type
  - Disadvantages: needs to be implemented on each machine and only detects after the intrusion
- Hybrid Intrusion Detection System
  - Advantages: much more effective
  - Disadvantages: huge implementation necessary, not privacy-friendly

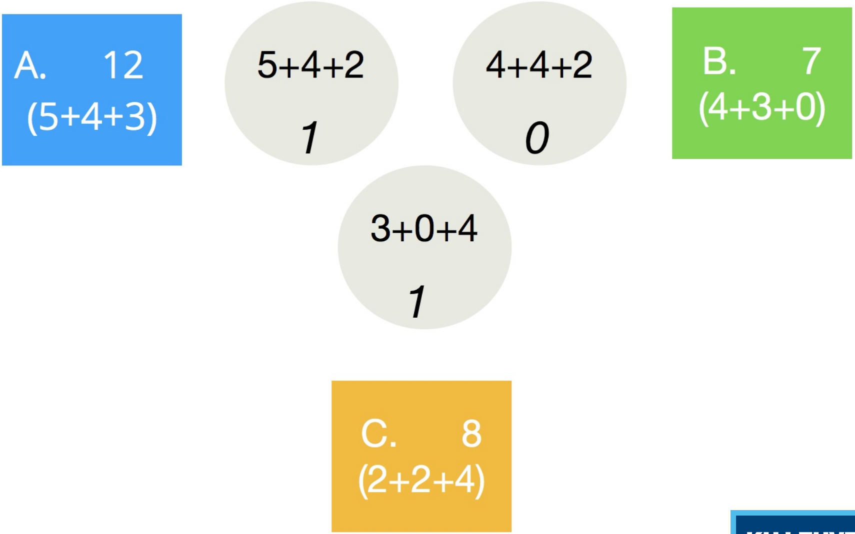
Privacy-friendly data pooling for machine learning network intrusion detection system

Addition over $\mathbb{Z}_2$

- i players have each a secret number n_i
- they want to know if the sum of their numbers is even or uneven. $\Sigma n_i \mod 2 = 0$ or 1 ?
- they don't want anybody except them to know their number

Solution

- each players divides its number n_i into j $m_{i,j}$ parts. $\Sigma m_{i,j} = n_i$
- j players each receive the $m_{i,j}$-part of each i players, sums it up and say if it is even or not. $\Sigma_i m_{i,j} \mod 2 = 0$ or 1.
- the results of all j players is then summed up and is even if $\Sigma n_i \mod 2 = 0$ and uneven otherwise. The problem is resolved.

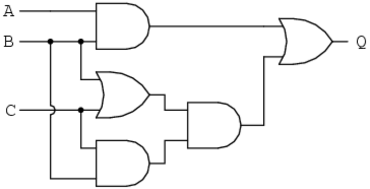

A. 12
(5+4+3)
5+4+2
1
4+4+2
0
B. 7
(4+3+0)
3+0+4
1
C. 8
(2+2+4)

Garbled

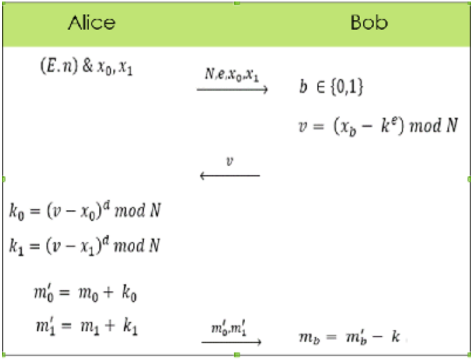
- Boolean circuit
- Based on OT
- Constant number of rounds

Arithmetic

- Based on somewhat homomorphic encryption (SHE)
- Allows for addition and multiplication



- The receiver doesn't get any information on the strings he didn't recieve
- The sender doesn't know which string was asked



g	e	output $g \wedge e$
0	0	0
0	1	0
1	0	0
1	1	1

$\Rightarrow$

garbled output
$\mathsf{Enc}(H(W_{\mathsf{G}}^0, W_{\mathsf{E}}^0), 0)$
$\mathsf{Enc}(H(W_{\mathsf{G}}^0, W_{\mathsf{E}}^1), 0)$
$\mathsf{Enc}(H(W_{\mathsf{G}}^1, W_{\mathsf{E}}^0), 0)$
$\mathsf{Enc}(H(W_{\mathsf{G}}^1, W_{\mathsf{E}}^1), 1)$

$\Rightarrow$

permuted garbled output
$\mathsf{Enc}(H(W_{\mathsf{G}}^0, W_{\mathsf{E}}^1), 0)$
$\mathsf{Enc}(H(W_{\mathsf{G}}^1, W_{\mathsf{E}}^1), 1)$
$\mathsf{Enc}(H(W_{\mathsf{G}}^0, W_{\mathsf{E}}^0), 0)$
$\mathsf{Enc}(H(W_{\mathsf{G}}^1, W_{\mathsf{E}}^0), 0)$

Fig. 1. The garbling of an AND gate

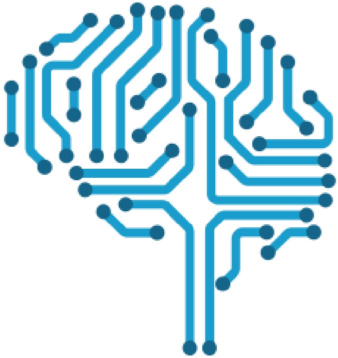
Different leads

- Fairplay (boolean, n-party)
- BMR (boolean, n-party)
- Sharemind (3-party, proprietary)
- VIFF (obsolete)
- ABY (2-party)
- **SPDZ (artihmetic, n-party)**

Privacy-friendly collaborative network intrusion detection system

Datasets

- KDD CUP '99
- AWID
- PCAP
- UNB ISCX
- CSIC 2010 HTTP Dataset
- West Point - NSA DataSet



Current research algorithms

- Semi-supervised learning algorithms with fuzziness
- K-Nearest Neighbors
- Support Vector Machines
- Bayes Classifier
- EM-Clustering
- Genetic algorithms
- Classification Tree

- Build on existing application (e.g. Bro Network Security Monitor, OpenNMS,...), in the form of a plug-in.
- Feed with constant new data, provided from other analysis tools
- User-friendly
- Off-line

- Begin March: Benchmark and designing working machine-learning algorithm
- Begin April: Design of final algorithm including MPC
- Begin May: Prototyping as a plug-in
- Begin June: Poster and report

[1] D. Catalano, R. Cramer, G. D. Crescenzo, I. Darmgård, D. Pointcheval, and T. Takagi, *Contemporary Cryptology*. Birkhäuser Basel, 2005.

[2] A. R. B. S. P. F. Atmaja Sahasrabuddhe, Sonali Naikade, "Survey on intrusion detection system using data mining techniques," *International Research Journal of Engineering and Technology (IRJET)*, may 2017.

Further optimizations

- Hybrid Intrusion Detection Systems
- Speed and complexity optimizations (research on HE)
- Deep learning


Any
Questions