Institute of Computer Science, University of Bern
Cryptographic Protocols, FS 2021

Prof. Christian Cachin
Jovana Mićić

Exercise 11

11.1 Arithmetic circuits (4pt)

The BGW protocol for secure multiparty computation evaluates an arithmetic circuit among n parties. However, most computer hardware, hardware-layout compilers, and the familiar circuit model in theory operate with binary (Boolean) wires and gates.

Describe how to turn a binary circuit into an equivalent arithmetic circuit over $GF(q)$, for some prime q. To be concrete, represent FALSE and TRUE with values in $GF(q)$ and describe how to implement AND, OR, NOT, and XOR gates. Justify the operations of each gate.

11.2 Multiplication gate with preprocessing (6pt)

Since the evaluation of multiplication gates in the BGW protocol is expensive, it is the first target for optimization. One practical way is described here and uses precomputed triples for those gates.

For each triple, random x and y in $GF(q)$ are generated collaboratively in a preprocessing step so that the parties hold $[x]$ and $[y]$. Then $[z] = [xy]$ is computed using the BGW protocol.

Later, during the evaluation of the actual circuit, the parties may compute the multiplication of $[w_j]$ and $[w_k]$ with simple operations only; it should result in a sharing of $[w_t] = [w_j w_k]$. (All computations here are in $GF(q)$.)

To start this protocol, the parties compute $[w_j - x]$ and $[w_k - y]$ locally and then reconstruct (using the protocol for output wires) the values $m_j = w_j - x$ and $m_k = w_k - y$ using the output-reconstruction protocol. (Notice that m_j and m_k are the wire values w_j and w_k *masked* by x and y, respectively.)

Prove that

$$w_j w_k \ = \ m_j m_k + m_j y + m_k x + z$$

and show how to evaluate $[w_t] = [w_j w_k]$ by stating and justifying the protocol steps.