REPORT 20/4/2011

****************

Secure function evaluation (SFE) paradigm
- introduced by Yao (1986)
- allow two (or more) parties, each with data $x^i = (x_1^i, ..., x_k^i)$, to evaluate a known function $f$ over all the data, i.e. compute $f(x^1, ..., x^n)$, without any party knowing any of the other parties' data
- in real life this can be done using a trusted party, but using cryptographic tools such party is not needed

Oblivious transfer
- allows one party (sender) to transmit partial data to another party (receiver) in a secure way
- the sender has the guarantee that the receiver only learns part of the original data
- the receiver has the guarantee that the sender does not know which part of the data he has learned

Garbled circuits
- consider functions to be implemented by sets of logic gates
- assign random (garbled) values to each 0/1 value of each wire
- build tables to compute garbled outputs of the logic gates with the garbled input values
- make sure that the tables only allow computing the garbled outputs, and nothing else

Fairplay system overview
- write SFDL program
- translate SFDL program to SHDL circuit
- parse SHDL circuit → Java object
- Bob constructs $m$ garbled circuits, sends them to Alice; Alice chooses one to be evaluated
- Bob exposes the other $m-1$ circuits; Alice verifies them against the circuit she chose
- Bob sends his garbled inputs to Alice; Alice inserts them in her circuit
- Alice sends her inputs to Bob using oblivious transfer, Bob returns her inputs in garbled form
- Alice evaluates her circuit and sends Bob his garbled outputs
- Both Alice and Bob decode their outputs