An EasyCrypt formalization of Garbled Circuits

How to prove a "compiler" in easycrypt

Guillaume Davy - IMDEA Software, ENS Cachan
- Joint work with :
José Carlos Bacelar Almeida - Universidade do Minho
Manuel Bernardo Barbosa - Universidade do Minho
Gilles Barthe - IMDEA Software
François Dupressoir - IMDEA Software
Pierre-Yves Strub - IMDEA Software

Agenda

1. From MPC to garbled circuit
2. Garbled circuit, definition and security
3. Proof

a) Introduction

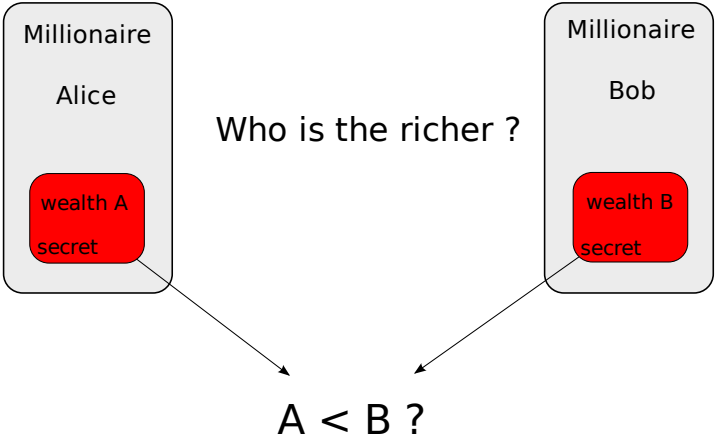
b) Multi Party Computation

c) Garbled circuit

1. From MPC to garbled circuit

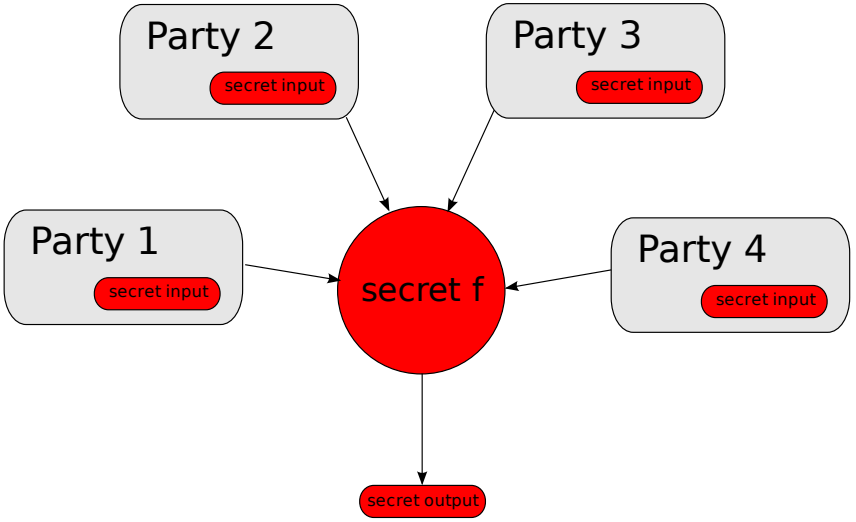
a) Multi Party Computation

Millionaires Problem :



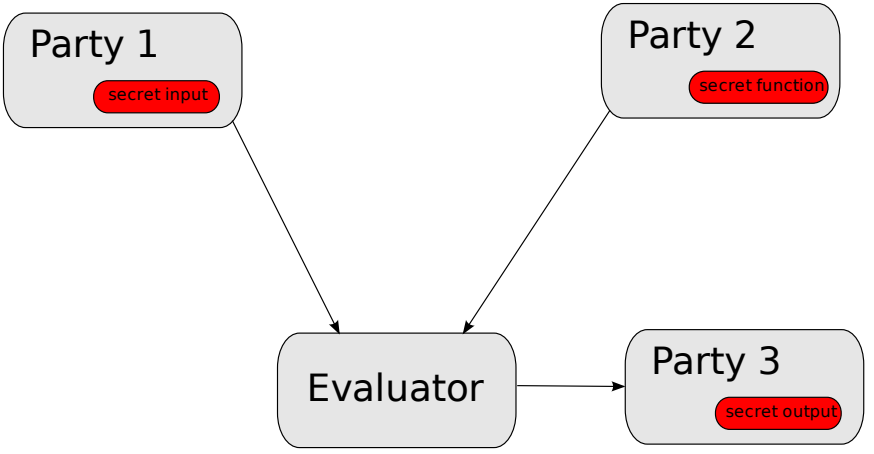
1. From MPC to garbled circuit

a) Multi Party Computation

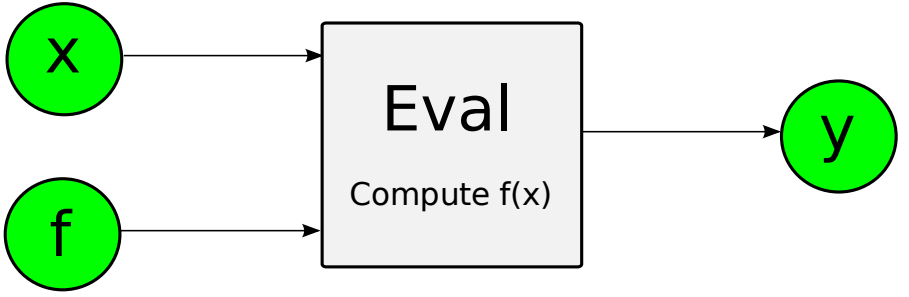


1. From MPC to garbled circuit

b) Secure Function Evaluation

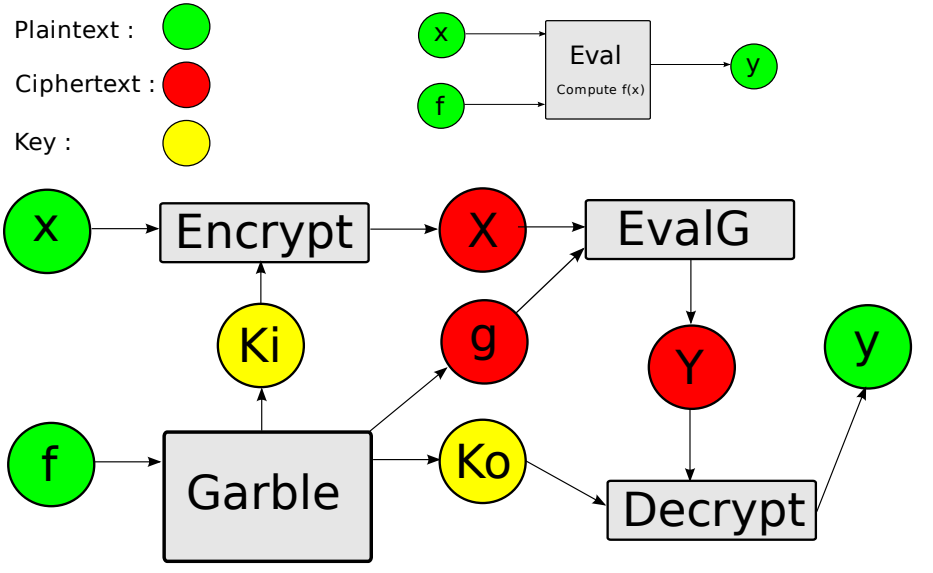


Evaluating a function on a particular input :



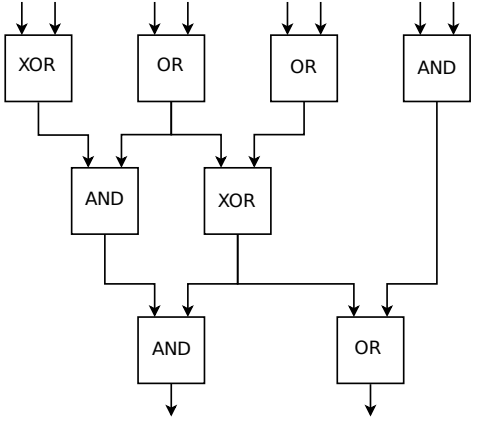
1. From MPC to garbled circuit

c) Garbled circuit



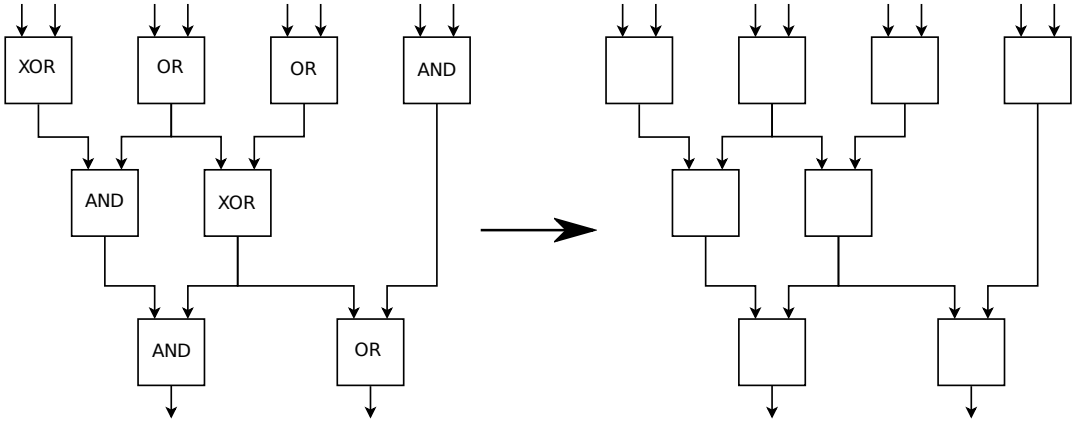
1. From MPC to garbled circuit

c) Garbled circuit



1. From MPC to garbled circuit

c) Garbled circuit



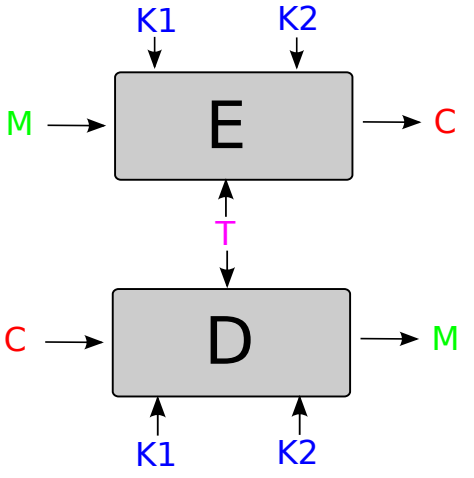
a) Dkc

b) Garbled circuit

c) Security notion

2. Garbled circuit, definition and security

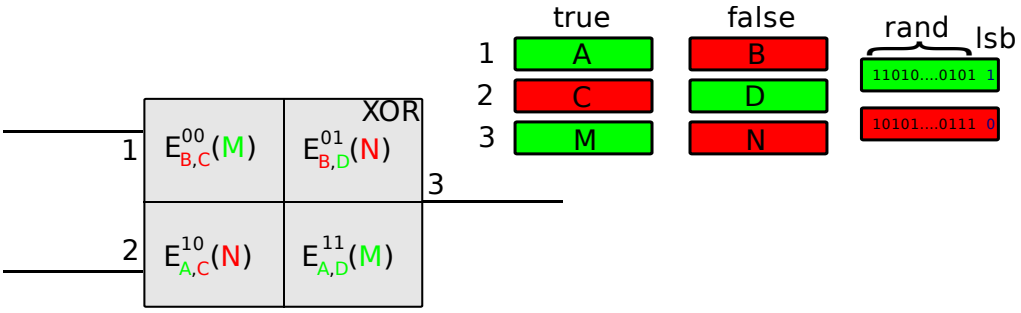
a) Dkc



$$E^{T}_{K1\,K2}(M)$$

2. Garbled circuit, definition and security

b) Garbled circuit

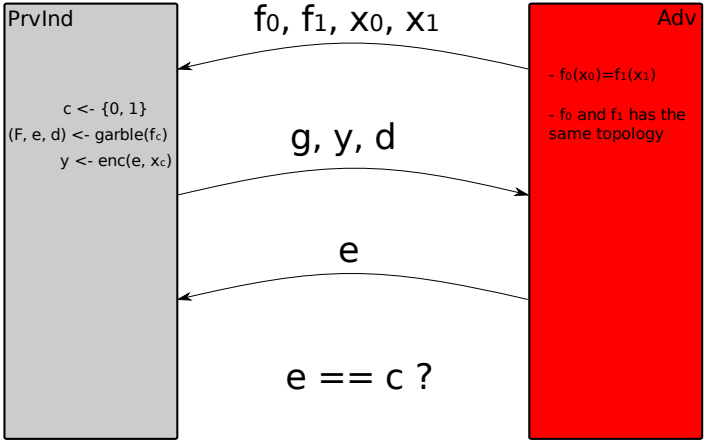


- For each wire we make two tokens one for false, one for true
- The input tokens of a gate are used to encrypt the output token via DKC
- The only way of knowing a token is to decrypt the output token of the corresponding gate which is possible only if you have the corresponing input
- If you are given the tokens for an input you will be able to compute the circuit on this input

2. Garbled circuit, definition and security

c) Security notion

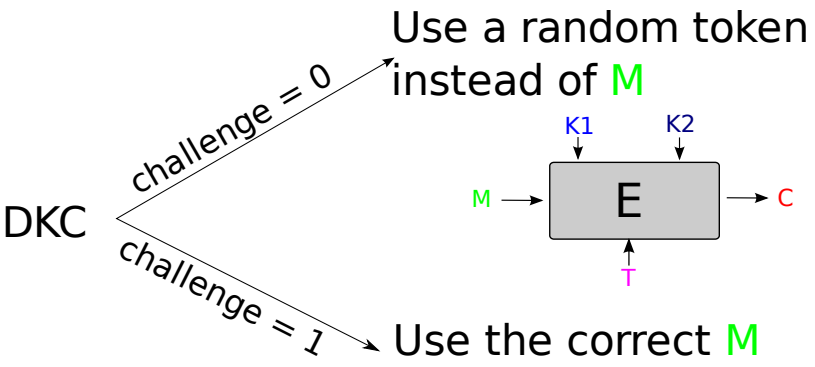
PrvInd :



2. Garbled circuit, definition and security

c) Security notion

DKC :



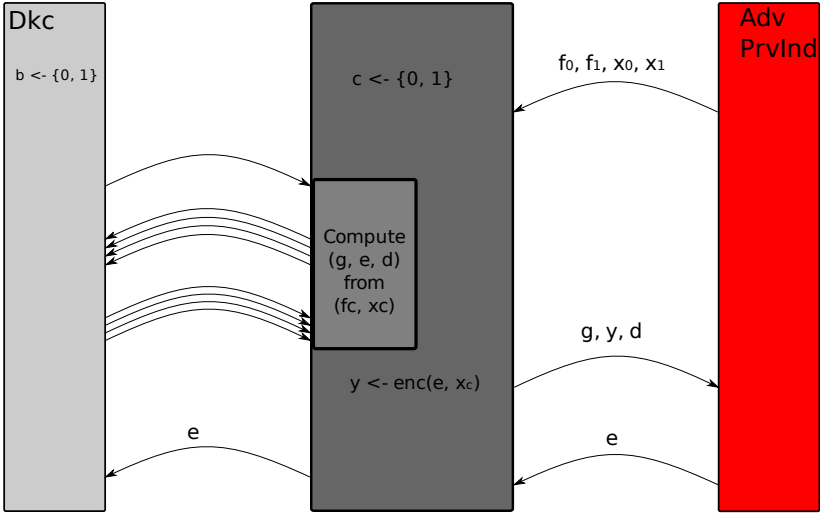
a) Reduction

b) Hybrid argument

c) Conclusion

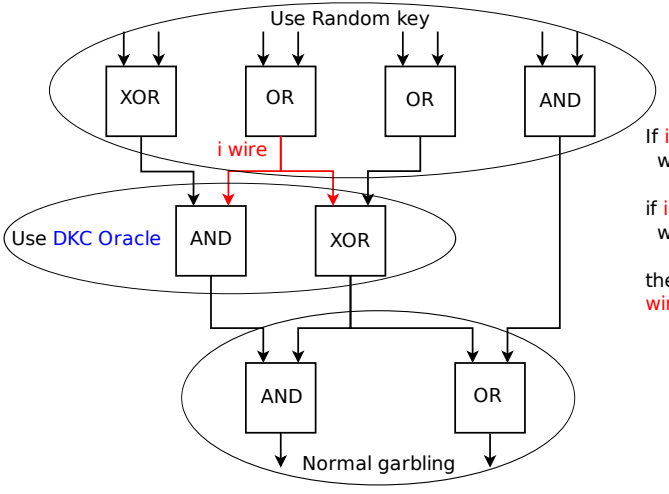
3. Proof

a) Reduction



3. Proof

b) Hybrid argument



If i = 0 and DKC is normal then
we have a normal garbling

if i = last wire and DKC is random then
we have a fake garbling

there is a relation between wire i and
wire i + 1

3. Proof

c) Conclusion

What is done :

- The formalization, of games and reduction
- The high level proof, that compute the probability
- 2 main equivalence proofs are almost done out of 3

What remains to be done :

- Finish the equivalence proof and do the last one (need eager)
- Some side conditions still have to be proved (e.g. losslessness of the reduction)
- Interaction with SFE and oblivious transfer

3. Proof

c) Conclusion

New easycrypt features useful for garbled circuit proof :

- Mean, Sum and Interval
- Abstract definition for security
- Hybrid argument
- Working with high order object : function

3. Proof

c) Conclusion

What's next ?

- Reuse definition and lemma in a higher level proof
- Transform easycrypt implementation into a concrete implementation that will be proved secure