Efficient Server-Aided Secure Two-Party Function Evaluation with Applications to Genomic Computation

Marina Blanton and Fattaneh Bayatbabolghani

Department of Computer Science and Engineering
University of Notre Dame

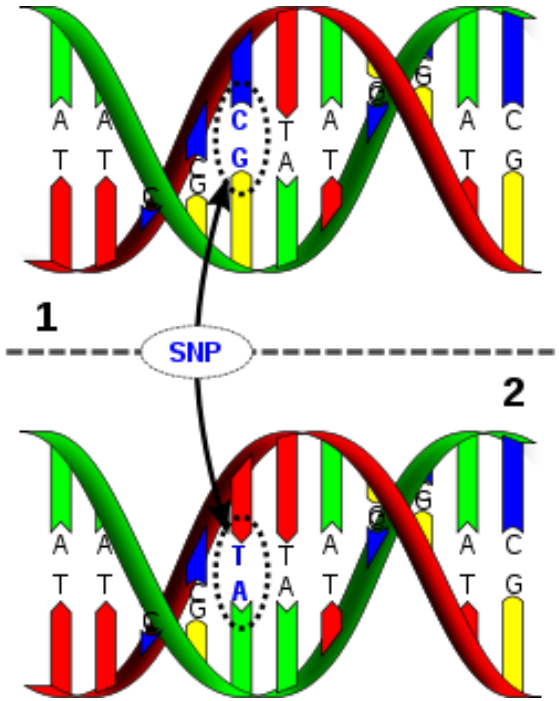
The 16th Privacy Enhancing Technologies Symposium
July 20, 2016

Motivation

- We treat the problem of securing computation associated with genomic tests used for voluntarily non-medical purposes
  - ✓ ancestry test
  - ✓ paternity test
  - ✓ genomic compatibility test
- There is a desire to protect highly sensitive DNA data of users participating in these tests
- Such computation often takes place between two users communicating via a server
  - ✓ this prompts the use of the server-aided secure two-party computation setting
  - ✓ participation of the server can significantly improve the participants' cost

Genomic Backgrounds

- Information extracted from one's genome is often represented in the form of SNPs or STRs
- Each SNP is referenced by a specific index and its value is 0, 1, or 2
- Each STR consists of a fixed number of pairs taking integer values



Genomic Tests

- **Ancestry test**
  - ✓ compares two SNP sequences belonging to two individuals
  - ✓ determines the number of SNPs they have in common
- **Paternity test with a single parent**
  - ✓ uses STR profiles S (the child) and S' (the contested father)
  - ✓ computes whether all elements of S and S' have at least one common component
- **Genetic compatibility test**
  - ✓ compares markers of potential partners
  - ✓ evaluates the possibility of their children inheriting genetic diseases

Security Model

- The computation takes the form of secure function evaluation with participating users A and B and server S
- We start by assuming that the server behaves **semi-honestly** and contributes **no input**
- Assumptions placed on the users vary
  - ✓ **semi-honest** users (ancestry test)
  - ✓ **malicious** users (paternity test)
  - ✓ **malicious** users **tampering with input** (genomic compatibility test)

Underlying Techniques

- We build solutions based on **garbled circuit evaluation** with different security guarantees
- Garbled circuits allows two parties P_1 and P_2 to securely evaluate a Boolean circuit of their choice
- A desired function should be converted to a Boolean circuit and then
  - ✓ one party acts as a **circuit generator**
  - ✓ the other party acts as a **circuit evaluator**
- The circuit generator creates two random labels for each (binary) wire of the circuit during garbling
- The evaluator sees only one label during evaluation and doesn't know its meaning

Underlying Techniques

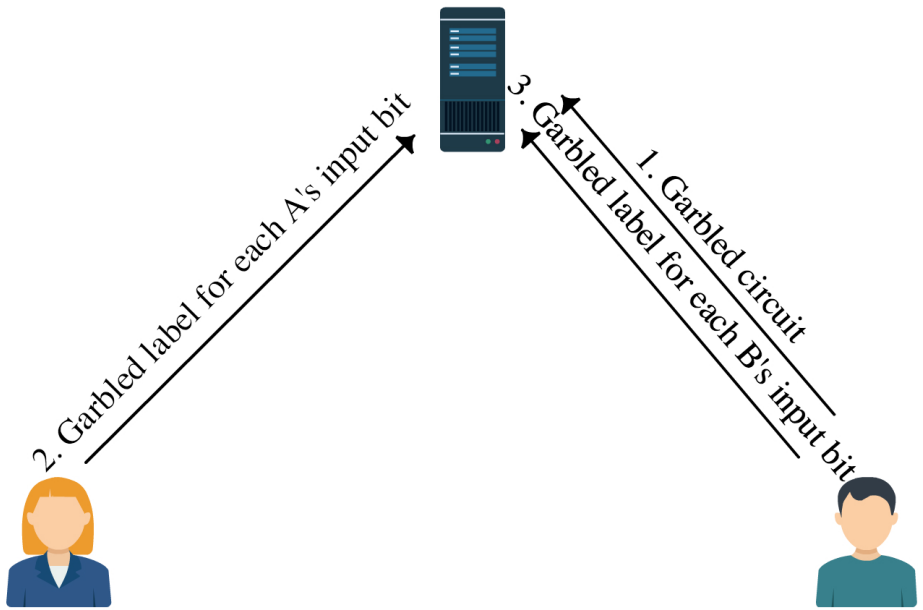
- The circuit generator directly sends to the evaluator the correct label corresponding to its input for each input wire
- The parties perform **oblivious transfer** (OT) to communicate labels corresponding to the circuit evaluator's input to the circuit evaluator
- In 1-out-of-2 oblivious transfer
  - ✓ the sender has inputs X_0 and X_1
  - ✓ the receiver obtains only X_b for its choice of bit b
  - ✓ the sender learns nothing

Scheme I: Semi-Honest Users, Malicious Server

- Our first construction
  - is cheaper than a regular garbled circuit protocol for two semi-honest parties
  - uses only garbled circuit generation and evaluation, but not OT
  - is intended for ancestry tests
  - security in the presence of a malicious server comes for free

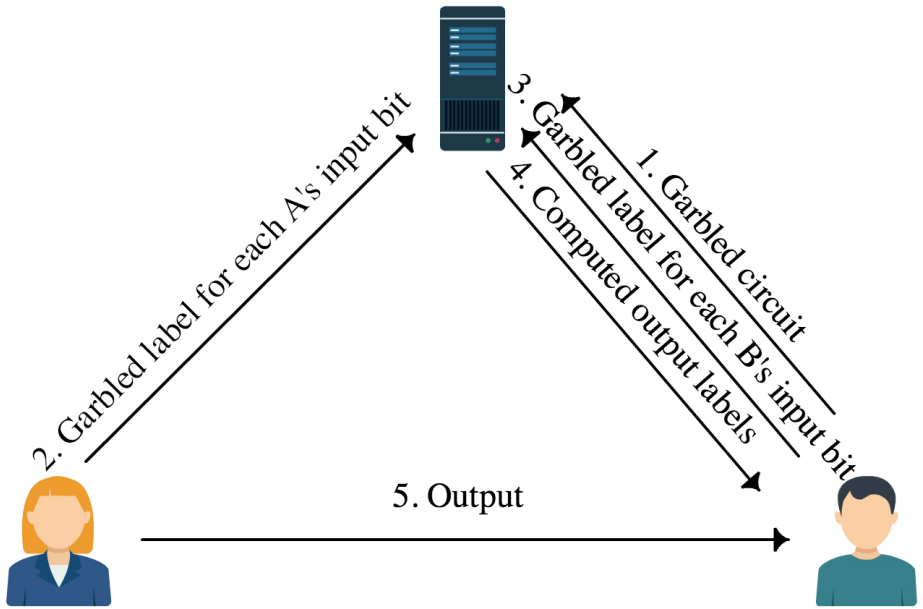
Scheme I: Semi-Honest Users, Malicious Server

- In our first construction:
  - A and B create a garbled circuit and send it to S (the task can be partitioned arbitrarily between them)
  - they also sent wire labels for their inputs to S



Scheme I: Semi-Honest Users, Malicious Server

- In our first construction:
  - S evaluates the garbled circuit and returns the output labels to A and B

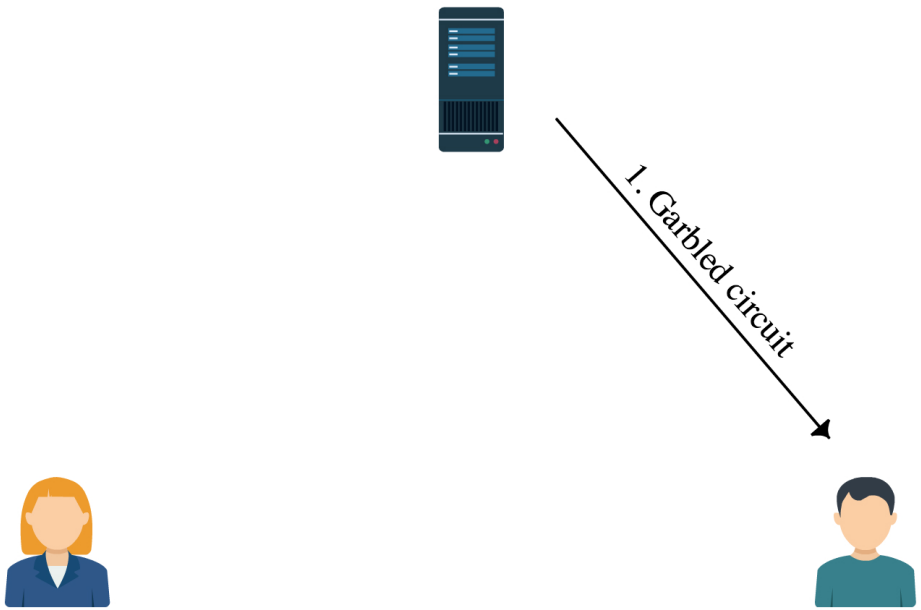


Scheme II: Malicious Users, Semi-Honest server

- Our second construction
  - has cost similar to a regular garbled circuit protocol for two semi-honest parties
  - is intended for paternity tests

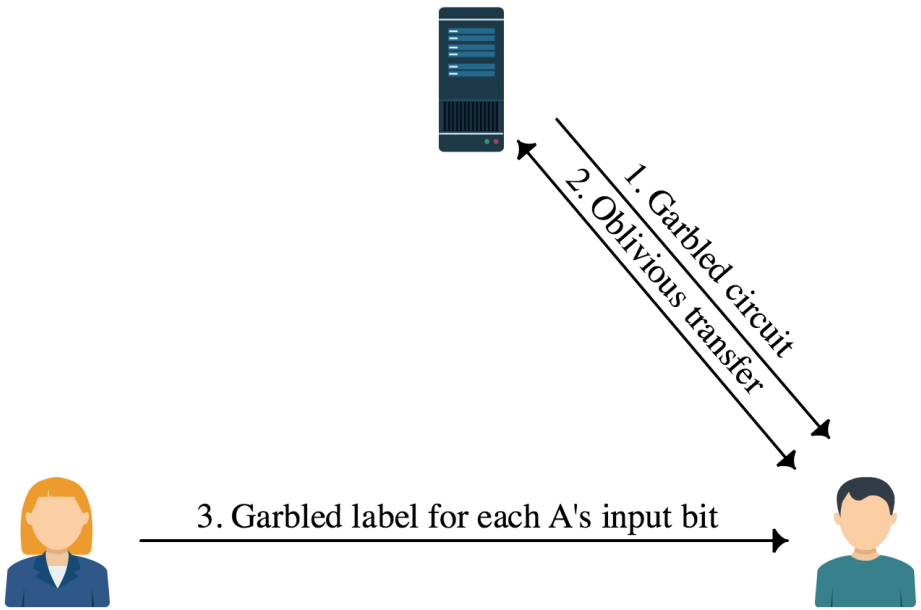
Scheme II: Malicious Users, Semi-Honest Server

- In our second construction:
  - S garbles the circuit and sends it to B
  - S communicates to A information about label pairs corresponding to A input wires



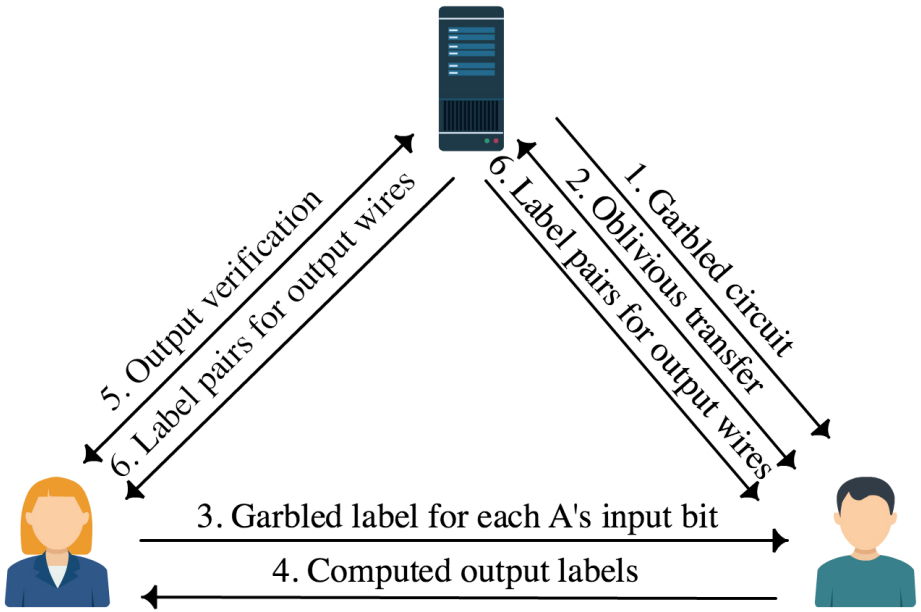
Scheme II: Malicious Users, Semi-Honest Server

- In our second construction:
  - B obtains wire labels for his input from S using OT and wire labels for A's input directly from A
  - B evaluates the circuit and sends the computed output wire labels to A



Scheme II: Malicious Users, Semi-Honest Server

- In our second construction:
  - A verifies validity of output wire labels obtained from B with the help of S
  - if A is satisfied, S opens the meaning of output wire labels to both A and B (**fairness**)



Scheme III: Malicious users, Semi-Honest Server, Input Certification

- We enhance Scheme II with input certification
  - A and B hold signatures (with special properties) on their inputs
  - A and B prove in zero knowledge possession of a signature and that the chosen input wire labels match the signed values
    - ✓ **this is the most interesting part from the technical point of view**
- The solution works for signatures on messages of any size, blocks of messages, a mix of certified and regular inputs, etc.

Performance Results

- **Ancestry testing**
  - each party inputs 2^{17} SNPs
  - the circuit used 655,304 XOR gates and 131,072 non-XOR gates
  - computation resembles the Hamming distance

	Garbled circuit		Communication	
	garble (offline)	eval (online)	sent	received
A	1.8ms	–	2MB	0MB
B	19.8ms	–	8MB	0MB
S	–	12.5ms	0MB	10MB

Performance Results

- **Paternity testing**
  - A and B input 13 pairs of STRs (x_i, y_i)
  - A is a parent of B if A's x_i or y_i appears in B's pair for each i
  - the circuit used 234 input bits from each party, had 468 XOR gates and 467 non-XOR gates

	Garbled circuit		OT		Total time		Communication	
	garble	eval	offline	online	offline	online	sent	received
A	0.003ms	–	–	–	0.003ms	–	3.7KB	0.06KB
B	–	0.01ms	515ms	202ms	515ms	202ms	31.7KB	56.9KB
S	0.03ms	–	196ms	261ms	196ms	261ms	53.3KB	31.7KB

Performance Results

- **Genetic compatibility testing**
  - A and B agree on k relevant genetic diseases and get their susceptibility to each of them tested and certified
  - the test outputs 1 if there is at least one disease to which both A and B are susceptible, and 0 otherwise
  - the circuit used $k = 10$ certified inputs for each party and had 19 non-XOR gates

	Garbled circuit		OT		Sign PK		Other PK		Total time		Comm	
	garble	eval	offline	online	offline	online	offline	online	offline	online	sent	received
A	0	–	–	–	1170	42	617	21	1790	63	34	0.06
B	–	0.001	15	15	1170	42	282	16	1470	72	36	3
S	0.003	–	29	15	0	2060	0	756	29	2830	3	71

  - computation times are in ms, communication is in KB, PK stands for proof of knowledge


Q&A