WHAT IS CLAIMED IS:

1. A method for performing a privacy-preserving multi-party computation, the method comprising performing, by a receiver device:

storing an input value b that is to be used in executing an encrypted program;

generating a first value r_{1-b} using a random number generator;

generating a key pair comprising a private key and a public key, using a cryptographic function;

generating a hash output using a hash function that corresponds to the cryptographic function;

generating a second value r_b using the public key and the hash output;

sending the first value r_{1-b} and the second value r_b to a sender device;

receiving a cryptographic value c from the sender device, wherein the cryptographic value c is generated using the cryptographic function;

generating, using the private key and the cryptographic function, a key value that corresponds to input value b;

receiving the encrypted program from the sender device;

executing the encrypted program using the key value to obtain an encrypted output value; and

sending the encrypted output value to the sender device, wherein the sender device decrypts the encrypted output value using the key value.

2. The method of claim 1, wherein the cryptographic value c is generated in parallel with the generation of the first value r_{1-b} and the second value r_b.

3. The method of claim 1, wherein the input value b is an element of a set of values associated with a number of possible keys.

4. The method of claim 1, wherein the hash function is a random oracle.

5. The method of claim 1, wherein the first value r_{1-b} has a same length in bits as the public key.

6. The method of claim 1, wherein generating the hash output using the hash function that corresponds to the cryptographic function comprises: executing the hash function on the first value r_{1-b}.

7. The method of claim 1, wherein generating, using the private key and the cryptographic function, the key value that corresponds to input value b comprises: using the cryptographic function and the private key to decrypt the cryptographic value *c* to produce the key value.

8. A method for performing a privacy-preserving multi-party computation, the method comprising performing, by a sender device:

storing a first key k_0 and a second key k_1;

receiving a first value r_0 and a second value r_1 from a receiver device;

generating a first public key using the first value r_0 and a first hash output by operating a hash function H on the second value r_1, wherein the hash function H correspond to a cryptographic function used by the receiver device to generate the first value r_0 and the second value r_1;

generating a second public key using the second value r_1 and a second hash output by operating the hash function H on the first value r_0;

generating a first cryptographic value c_0 by operating the cryptographic function on the first public key and the first key k_0;

generating a second cryptographic value c_1 by operating the cryptographic function on the second public key and the second key k_1;

sending the first cryptographic value c_0 and the second cryptographic value c_1 to the receiver device;

sending an encrypted program to the receiver device, wherein the receiver device executes the encrypted program to generate an encrypted output value;

receiving the encrypted output value; and

decrypting the encrypted output value using a corresponding key.

9. The method of claim 8, wherein the first value r_0 is randomly generated by the receiver device and the second value r_1 is generated by the receiver device using a public key and a hash output of executing the hash function H on the first value r_0.

10. The method of claim 8, wherein the receiver device is only able to decrypt one of the first cryptographic value c_0 or the second cryptographic value c_1, based on a secret key stored on the receiver device.

11. The method of claim 8, wherein the encrypted program is generated by the sender device.

12. The method of claim 11, wherein the encrypted program is keyed to either the first key k_0 or the second key k_1.

13. The method of claim 8, wherein the encrypted program is a garbled circuit.

14. A method for performing a privacy-preserving multi-party computation, the method comprising performing, by a sender device:

generating a first private key d using a random number generator;

generating a first public key c using a cryptographic function;

sending the first public key c to a receiver device;

receiving a first value r_0 and a second value r_1 from the receiver device;

generating a first key k_0 using the first value r_0 and a first hash output by operating a hash function H on the second value r_1 to obtain a first intermediate result, wherein the cryptographic function operates on the first intermediate result using the first private key d to generate the first key k_0, wherein the hash function H corresponds to the cryptographic function;

generating a second key k_1 using the second value r_1 and a second hash output by operating the hash function H on the first value r_0 to obtain a second intermediate result, wherein the cryptographic function operates on the second intermediate result using the first private key d to generate the second key k_1;

sending an encrypted program to the receiver device, wherein the receiver device executes the encrypted program to generate an encrypted output value;

receiving the encrypted output value; and

decrypting the encrypted output value using a corresponding key.

15. The method of claim 14, wherein generating the first public key c using the cryptographic function comprises: raising a value *g* to a power of *d* to produce the first public key *c*.

16. The method of claim 14, wherein the first value r_0 is randomly generated by the receiver device using a random number generator.

17. The method of claim 16, wherein the second value r_1 is generated by the receiver device, by subtracting a hash output of executing the hash function H on the first value r_0 from a value *g* raised to a power of a randomly generated number *a*.

18. The method of claim 14, wherein the encrypted program is keyed to either the first value r_0 or the second value r_1.

19. The method of claim 14, wherein the public key *c* raised to a power of a randomly generated number *a* is equal to one of the first key k_0 or the second key k_1.

20. The method of claim 14, wherein the hash function H is a random oracle.

21. A receiver device comprising:

one or more processors; and

a computer readable medium coupled to the one or more processor and containing instructions for causing the one or more processors to perform the method of any one of claims 1-7.

22. A sender device comprising:

one or more processors; and

a computer readable medium coupled to the one or more processors and containing instructions for causing the one or more processors to perform the method of any one of claims 8-20.

~~23. A computer readable medium containing instructions for causing the processor to perform the method of any one of claims 1-~~

WHAT IS CLAIMED IS:

1. A method for performing a privacy-preserving multi-party computation, the method comprising performing, by a receiver device:

storing an input value b that is to be used in executing an encrypted program;

generating a first value r_{1-b} using a first random number generator;

generating a second value a using a second random number generator;

generating a hash output using a hash function of the first value r_{1-b};

generating a first public key corresponding to the second value a using a cryptographic function;

generating a third value r_b using the first public key and the hash output;

sending the first value r_{1-b} and the third value r_b to a sender device;

receiving a second public key c from the sender device, wherein the second public key c is generated using the cryptographic function;

generating, using the second public key c and the second value a, a key value that corresponds to the input value b;

receiving the encrypted program from the sender device;

executing the encrypted program using the key value to obtain an encrypted output value; and

sending the encrypted output value to the sender device, wherein the sender device decrypts the encrypted output value using the key value.

2. The method of claim 1, wherein the input value b is an element of a set of values associated with a number of possible keys.

3. The method of claim 1, wherein the hash function is a random oracle.

4. The method of claim 1, wherein the first value r_{1-b} has a same length in bits as the first public key.

5. The method of claim 1, wherein the hash output has a same length in bits as the first public key.

6. The method of claim 1, wherein generating the key value comprises raising the second public key c to a power of the second value a.

7. The method of claim 1, wherein generating the first public key using the cryptographic function comprises raising a value g to a power of the second value a to produce the first public key.

8. The method of claim 1, wherein a private key d used to generate the second public key c is generated by using the second random number generator.

9. The method of claim 8, wherein generating the second public key c using the cryptographic function comprises raising a value g to a power of the private key d to produce the second public key c.

10. A receiver device comprising:

one or more processors; and

a computer readable medium coupled to the one or more processors and containing instructions for causing the one or more processors to perform the method of any one of claims 1-9.