Demonstrating that a Public Predicate can be Satisfied Without Revealing Any Information About How

David Chaum

Centre for Mathematics and Computer Science
Kruislaan 413 1098 SJ Amsterdam the Netherlands

It's not unlike a technique of probabilistic mathematical proof
in which you allow a receiver to select one of two cases.
—Norman Shapiro

[responding] *Yes, you're right....*
but the residue of doubt is provably, negligibly small.
—Michael Rabin 1977

Introduction

The problem solved here may be defined in the following way: Both parties y and z agree on a Boolean expression called a predicate; y claims to know a secret value satisfying the predicate; z wants very high certainty that y does have such a value; while y is willing to demonstrate possession of the secret satisfying value, y is unwilling to reveal the secret value to z. The solution requires z to assume that y cannot quickly solve certain problem instances provided by z. But y is sure not to reveal anything about the secret, even if z has unlimited computing power.

Relation to Other Work

The result presented is a dual of those by [Goldreich, et al 86] and [Brassard & Crepeau 86]: their model is an x with infinite computational ability and a z with limited ability; here z may have infinite computational ability and y has only limited ability. Besides being of theoretical interest for this reason, the approach presented here offers several advantages:

- The only possibility for cheating is to solve specific instances of the hard problem (factoring in the example construction) within the time allotted to compute legal responses.
- A variation is secure even if some known fraction of instances of the assumed hard problem

A.M. Odlyzko (Ed.): Advances in Cryptology - CRYPTO '86, LNCS 263, pp. 195-199, 1987.

can be solved within the allotted time.

- If there are multiple solutions, no information about which one(s) the prover knows is released by the protocol, even to someone who actually has infinite computing power.
- The model is consistent with previous proposals of the author [Chaum 85b], where an individual may have to demonstrate something to an organization that has potentially unknown resources or abilities. In fact, the result is a special case of a protocol previously presented by the author [Chaum 85a], whose properties are described in [Chaum 85b page 1039]. But the underlying problem assumed hard in that work differs from those relied on here.
- Giving the verifier a chance to cheat of less than 2^s requires only an amount of computation linear in s and the number of gates needed to represent the predicate. For $s = 100$ and say 200 digit composites, this requires for each gate only about as much computation as a single RSA decryption.
- The protocol is easily adapted to the dual model.

1. PROTOCOL

In overview, the protocol presented involves y making known to z transformed and encrypted copies of a truth table for each gate of a circuit representation of the predicate, after which z is allowed to "select one of two cases". The basic idea of getting exponential security by one party first committing by revealing encrypted forms and then allowing the other party to choose between several cases, which is relied on here, was first proposed in the context of cryptographic protocols by Rabin in [77] (which is the subject of the discussion quoted at the beginning of this article).

1.1 Protocol Set-Up

Initially y and z agree on a predicate and its realization by a circuit comprising m gates $g_1, \ldots, g_m$, defined by their respective truth tables $T_1, \ldots, T_m$. The gates are interconnected by n wires $w_1, \ldots, w_n$, with each column of every truth table corresponding to a wire. Thus the predicate may be thought of as a Boolean function on say r secret input bits involving m elementary Boolean operations each (except one) of whose output bits becomes an input for one or more other elementary operations without feedback. This means that the memoryless circuit has r input wires, each of which is an input to one or more gates (elementary operations defined by a corresponding truth table); $n - r - 1$ internal wires, each serving as the output of a single gate and "fanning-out" to serve as input to one or more other gates; and a single output wire of a single gate, which is the output of the whole circuit.

Consider a gate g_k with l inputs and an output defined by a truth table T_k (subsequently denoted without subscript) represented in matrix form as $T = (t_{i,j})$, with $i \in \{1, \ldots, 2^l\}$ and $j \in W_k$, where W_k is the set of wires corresponding to the inputs and outputs of gate g_k and (the cardinality) $\# W_k = l+1$, which is the total number of inputs and outputs of gate g_k, and $W_k \subset \{w_1, \cdots, w_n\}$. The entries of T are 0's and 1's, i.e. $t_{i,j} \in \{0,1\}$, in the usual way: the rows (apart from the last column) contain all defined input configurations, and the last entry in each row is the corresponding output.

It is sufficient to consider all the wires as having secret values, except the single output wire for the whole predicate. Since the value of this wire should be 1, the truth table of its gate is modified as follows: all rows with 0 in the output column are removed, and then the output column itself is removed.

First, y choses an inversion I_j at random for each wire w_j, i.e. $I_j \in \{0,1\}$ for $j \in \{w_1, \ldots, w_n\}$, where random choices (as used throughout) are uniform choices that are statistically independent of everything else.

Next, y successively transforms each T, first to a permuted form T', second to an obscured form T'', and third to an encrypted form E as follows: (a) Each T is transformed into a matrix $T' = (t'_{i,j})$, by a random row permutation. (b) Each T' is transformed into a table $T'' = (t''_{i,j})$ for which all entries in all columns corresponding to inverted wires are inverted: $t''_{i,j} = t'_{i,j} \oplus I_j$. (c) Each entry of the obscured form T'' is encrypted in a special way to yield $E = (e_{i,j})$: for each entry in T'' a random residue modulo N that is coprime with N, shown as $r_{i,j}$, is chosen with Jacobi symbol $(r_{i,j} / N)$ equal 1 when $t''_{i,j} = 1$ and equal -1 otherwise, and $e_{i,j} \equiv r_{i,j}^2 \pmod{N}$, where N is supplied to y by z.

Then y displays all the matrices E to z and allows z to choose between two cases:

(1) Display by y of I_j and, for each gate, all the $r_{i,j}$'s used in forming the corresponding E's. This allows z to recover every T'' from the Jacobi symbols of the $r_{i,j}$'s, to check that the entries of each E are the squares of the corresponding $r_{i,j}$, and to verify that each T'' satisfies $t''_{j,i} = t'_{i,j} \oplus I_j$, for some row permutation T' of T.

(2) Display by y of one row of $r_{i,j}$'s for each E, which should correspond to the actual row of the truth table that is satisfied by the secret wire values. This allows z to check that the entries of a row of each E are the squares of the corresponding r's, to recover the corresponding rows of the T'''s from the Jacobi symbols of the $r_{i,j}$'s, and to verify that all entries $t''_{i,j}$ of the displayed rows with the same j are equal.

2. SECURITY

Theorem: *No Shannon-information about the secret wire values is revealed by* y *following the protocol, assuming* N *has only two odd prime factors and they are each congruent to 3 modulo 4.*

Proof: First note that no information in the Shannon sense is revealed before z chooses a case, since each quadratic residue displayed has exactly the same probability of corresponding to a 1 as to a 0, because it has exactly two distinct roots with each Jacobi symbol. The secret wire values have no influence on what is revealed in case 1. In case 2, the indices of the displayed rows reveal nothing since the permutation of rows is chosen at random; a bit with index j in a revealed row corresponds with the jth wire, is equal to all other such bits with index j, and is just the exclusive-or of the secret wire value with I_j, which is just the encryption of the secret value under a true one-time pad.□

Theorem: *The probability that* y *satisfies* z*'s verification cannot exceed ½ when* y *is unable to learn secret wire values satisfying the circuit, assuming* y *cannot find two square roots of the same residue modulo* N *that have distinct Jacobi symbols.*

Proof: It is sufficient to show that if y can satisfy z in both cases, then y can learn wire values satisfying the circuit. All T'' are uniquely determined (from the assumption), are known to y, and contain only valid truth table rows when exclusive-ored with the corresponding bits of the I_j's known to y, as a consequence of y being able to satisfy case 1. From case 2, y knows a way to choose one row from each table T'' such that each wire is assigned the same value in all the chosen rows. Thus, y can form the exclusive-or of the I_j's known from case 1 with the rows known from case 2, which yields a valid row for each gate (from case 1) with an assignment of bits to wires that satisfies each such row (from case 2).□

Lemma: *If the above protocol is successfully repeated* s *times, using moduli each of which can be factored in the allotted time with independent probability p, then the probability of one-half in the previous theorem may be replaced by* $(½ + p / 2)^s$.

Proof: Follows immediately from elementary probability theory.

3. DISCUSSION

The protocol description used certain well known number theoretic functions (first introduced by Blum [82]) for clarity and concreteness, but the present results should not be interpreted as limited to these specific functions. A natural generalization is to any pair of so called "claw free" (as defined in [Goldwasser et al 85]) one-way bijections with the same image. Other choices of encryption functions switch the protocol to the dual model mentioned in the introduction: any suitable encryption of a single bit (or actually row of bits) with a unique inverse message could be used to encrypt a T'' to form an E.

In the protocol presented above, y must be convinced that N is a "Blum integer," or better, that it is of the form used in [Goldwasser et al 85]. There are at least two ways to address such a requirement. One is just to complete the protocol and then let y reveal the factorization of N to convince z that no cheating has occurred. When such an after-the-fact check is not acceptable, and where the particular encryption functions used require some such checking based on trapdoor information, z could use a protocol of the dual type to convince y that a predicate indicating suitability of the functions is satisfied.

Other claw free functions based on the discrete log problem do not require such checking [Damgård 86].

Acknowledgements

I am pleased to thank Oded Goldreich and Silvio Micali for their excitement about the difference between this work and their own and for encouraging me to publish it. Leoned Levin also expressed enthusiasm and reminded me about the claw free property. Additionally, thanks to Adi Shamir and Johan Hastad for listening to various versions of the protocol and inspiring its simplification; to Jeroen van de Graaf for several discussions; and to Jan-Hendrik Evertse for his comments.

References

(1) Blum, M., "Coin flipping by telephone," Proceedings of IEEE Compcon, 1982, pp. 133-137.

(2) Brassard, G. and Crepeau, C., "Zero-knowledge simulation of boolean circuits," preprint of extended abstract, April 1986.

(3) Chaum, D., "Showing credentials without identification: signatures transferred between unconditionally unlinkable pseudonyms," Presented at Eurocrypt'85, Linz Austria, April 1985a.

(4) Chaum, D., "Security without identification: transaction systems to make big brother obsolete," *Comm. ACM* 28, 10 (October 1985b), pp. 1030-1044.

(5) Damgård, I., private communication 1986.

(6) Goldreich, O., Micali, S., and Wigderson, A., "Proofs that yield nothing but the validity of the assertion and the methodology of cryptographic protocol design," preprint, April 1986.

(7) Goldwasser, S., Micali, S. and Rivest, R.L., "A 'paradoxical' solution to the signature problem," FOCS 84.

(8) Rabin, M.O., "Digitalized signatures," in *Foundations of Secure Computation*, Academic Press, NY, 1978.