

## Joint Encryption and Message-Efficient Secure Computation

Matthew Franklin\*

AT&T Bell Laboratories, 600 Mountain Avenue,  
Murray Hill, NJ 07974-0636, U.S.A.  
franklin@research.att.com

Stuart Haber

Bellcore, 445 South Street,  
Morristown, NJ 07960-6438, U.S.A.  
stuart@bellcore.com

Communicated by Joan Feigenbaum

Received 11 April 1994 and revised 27 January 1995

**Abstract.** This paper addresses the message complexity of secure computation in the (passive adversary) privacy setting. We show that  $O(nC)$  encrypted bits of communication suffice for  $n$  parties to evaluate any boolean circuit of size  $C$  privately, under a specific cryptographic assumption. This work establishes a connection between secure distributed computation and group-oriented cryptography, i.e., cryptographic methods in which subsets of individuals can act jointly as single agents. Our secure computation protocol relies on a new group-oriented probabilistic public-key encryption scheme with useful algebraic properties.

**Key words.** Secure computation, Privacy, Group-oriented cryptography, Distributed computing, Communication complexity.

### 1. Introduction

This paper connects two areas of recent cryptographic research: secure distributed computation and group-oriented cryptography. Several solutions have been found to the problem of securely evaluating an arbitrary boolean circuit under cryptographic assumptions, beginning with the work of Yao [14] and Goldreich *et al.* [7]. The notion of group-oriented cryptography, in which the power of a secret key holder is distributed over a number of participants, was introduced by Desmedt [2].

We focus on the message complexity (i.e., number of encrypted bits of communica-

---

\* Work performed while at Columbia University, with the support of a summer internship at Bellcore and a visiting position at C.W.I.

tion) of secure computation protocols in the privacy setting. Informally, a protocol is private if no subset of “gossiping” (but otherwise honest) participants can extract useful information about the nongossipers’ inputs. All previous methods for private computation required a message complexity that was quadratic in the number of participants. In this paper we show an improvement to a message complexity that is linear in the number of participants, under a specific cryptographic assumption.

### 1.1. *Previous Results in Message-Efficient Secure Computation*

Previously, the lowest message complexity known for  $n$  parties to evaluate a circuit of size  $C$  privately under reasonable cryptographic assumptions was  $O(n^2C)$  encrypted bits of communication. This same complexity was achieved using either of the main techniques for secure circuit evaluation in the cryptographic setting: the “gate-by-gate” approach or the “circuit-scrambling” approach.

In the gate-by-gate approach, each gate of the circuit is computed by having each pair of the  $n$  parties perform a private two-party protocol. In the protocol of Galil *et al.* [6], with efficiency improvements by Goldreich and Vainish [8], each two-party protocol is a single instance of “One out of Two Oblivious Transfer” (1-2-OT). It is possible to implement two-party 1-2-OT privately using a constant number of encrypted bits under a cryptographic assumption (e.g., three encrypted bits suffice under the assumption that composite quadratic character is hard to compute). This gives a total message complexity of  $O(n^2C)$  encrypted bits.

In the circuit-scrambling approach, each party takes a turn in modifying the truth tables of the gates of the circuit. In the protocol of Chaum *et al.* [1], each party can randomly permute the rows, and can randomly complement certain rows and columns of each truth table. Records of each party’s modifications are preserved in the form of bit commitments, which accompany the scrambled circuit as it passes from party to party (to enable circuit evaluation after the  $n$ th party has finished scrambling). Each party contributes a constant number of bit commitments for each gate (e.g., one bit commitment for each truth table row), and so the scrambled circuit as it passes from party  $i$  to party  $i + 1$  includes  $O(iC)$  bit commitments. When each bit commitment is a single encryption, this gives a total message complexity of  $O(n^2C)$  encrypted bits. These results rely on a bit commitment scheme with special properties, and can be based, e.g., on the intractability of computing discrete logarithms or composite quadratic character.

### 1.2. *Contributions of this Paper*

Our methods make novel use of ideas from group-oriented cryptography. In a group-oriented cryptosystem, any subset of the group can “participate” in an encryption. The public keys of the participants are used to encrypt a message, and the cooperation of all participants is needed to decrypt it. Practical implementations of group-oriented public-key encryption were given by Desmedt and Frankel [3]. (See also the related notion of fair public-key encryption [11].) We extend their implementations to achieve a “joint” encryption scheme with additional desirable properties:

- (Compact) The size of a jointly encrypted bit is independent of the number of participants.

- (Xor-Homomorphic) It is easy for anyone to compute a joint encryption of the XOR of two jointly encrypted bits.
- (Blindable) It is easy for anyone to “disguise” a joint encryption by replacing it with a random ciphertext that decrypts to the same message.
- (Witnessable) It is easy for any participant to “withdraw” from a joint encryption noninteractively (i.e., by sending a single value to the other participants).

Using this new scheme, we can reduce by a factor of  $n$  the number of bits broadcast by  $n$  parties to compute a circuit of size  $C$  securely. Specifically, in the privacy setting (against a passive adversary), only  $O(nC)$  encrypted bits of communication are needed.

We give our model and background notions in Section 2, our new joint encryption scheme in Section 3, our new protocol for message-efficient secure computation in Section 4, and some concluding remarks in Section 5.

## 2. Model and Background Notions

In this section we give some definitions and facts about distinguishability of distributions [13], [9], and then present our model for secure computation.

### 2.1. Distinguishability of Distributions

Two families of distributions  $\{\Delta_k^1\}$ ,  $\{\Delta_k^2\}$  are said to be “computationally indistinguishable” if no probabilistic polynomial-time (p.p.t.) Turing machine “distinguisher” can decide which distribution it is sampling from with a probability of success nonnegligibly better than random guessing. More formally, for every p.p.t. Turing machine  $M$  and every  $c, m > 0$ , there is  $K > 0$  such that, for all  $k > K$ ,  $|p_k^1 - p_k^2| < k^{-c}$ , where  $p_k^i = \text{prob}(M[d_1, \dots, d_m] = 1: d_1, \dots, d_m \leftarrow \Delta_k^i)$ .

Two families of distributions  $\{\Delta_k^1\}$ ,  $\{\Delta_k^2\}$  are said to be “statistically indistinguishable” if the distance between distributions is negligible. More formally, for all  $c > 0$  there is a  $K > 0$  such that, for all  $k > K$ ,  $k^{-c} > |\Delta_k^1 - \Delta_k^2| = \sum |\Delta_k^1[x] - \Delta_k^2[x]|$ , where the sum is over the domain of the distributions.

When the parameter  $k$  for a family of distributions is obvious, we often omit it, with a corresponding extension of terminology. For example, we may call two *distributions* indistinguishable when the implied families are indistinguishable. Throughout this paper the parameter  $k$  for a family of distributions is always assumed to be the security parameter (also denoted  $k$ ) for our joint encryption scheme.

Here are a few facts about families of distributions that will be useful. The first is that indistinguishability is transitive.

**Fact 1.** *If  $\Delta_1, \Delta_2$  are computationally (statistically) indistinguishable and  $\Delta_2, \Delta_3$  are computationally (statistically) indistinguishable, then  $\Delta_1, \Delta_3$  are computationally (statistically) indistinguishable.*

**Proof.** When  $\Delta_1, \Delta_2$  and  $\Delta_2, \Delta_3$  are statistically indistinguishable, the proof follows from the triangle inequality:  $|\Delta_1 - \Delta_3| = \sum_z |\Delta_1[z] - \Delta_3[z]| \leq \sum_z |\Delta_1[z] - \Delta_2[z]| + |\Delta_2[z] - \Delta_3[z]| = |\Delta_1 - \Delta_2| + |\Delta_2 - \Delta_3|$ . When  $\Delta_1, \Delta_2$  and  $\Delta_2, \Delta_3$  are computationally

indistinguishable, then a distinguisher  $M$  for  $\Delta_1, \Delta_3$  must also be a distinguisher either for  $\Delta_1, \Delta_2$  or for  $\Delta_2, \Delta_3$ , again by the triangle inequality:  $|p_k^1 - p_k^3| \leq |p_k^1 - p_k^2| + |p_k^2 - p_k^3|$ , where  $p_k^i = \text{prob}(M[d_1, \dots, d_m] = 1: d_1, \dots, d_m \leftarrow \Delta_k^i)$ ; thus at least one of the terms on the right-hand side of the inequality is nonnegligible whenever the left-hand side is nonnegligible.  $\square$

The second useful fact is that indistinguishability is preserved under products. The product distribution  $\Delta_1 \times \Delta_2 \times \dots \times \Delta_m$  is defined to be a distribution on  $m$ -tuples where the  $i$ th component is sampled according to the distribution  $\Delta_i$ .

**Fact 2.** *If  $\Delta_1, \Delta_2$  are statistically (computationally) indistinguishable, then  $\Delta_1 \times \Delta$  and  $\Delta_2 \times \Delta$  are statistically (computationally) indistinguishable, for any independent  $\Delta$  (that is samplable in probabilistic polynomial time).*

**Proof.**  $|\Delta_1 \times \Delta - \Delta_2 \times \Delta| = \sum_{y,z} |(\Delta_1 \times \Delta)[y,z] - (\Delta_2 \times \Delta)[y,z]| = \sum_z \Delta[z] \sum_y |\Delta_1[y] - \Delta_2[y]| \leq \sum_y |\Delta_1[y] - \Delta_2[y]|$ , which is negligible when  $\Delta_1, \Delta_2$  are statistically indistinguishable, implying the statistical indistinguishability of  $\Delta_1 \times \Delta, \Delta_2 \times \Delta$ . For the other case, we argue the contrapositive, i.e., that the computational distinguishability of  $\Delta_1 \times \Delta, \Delta_2 \times \Delta$  implies the computational distinguishability of  $\Delta_1, \Delta_2$ . If  $\Delta_1 \times \Delta, \Delta_2 \times \Delta$  have a p.p.t. distinguisher  $M$ , then consider the following machine  $M'$ : Given  $m = \text{poly}(k)$  samples from  $\Delta_i$  as input,  $M'$  constructs  $m$  samples from  $\Delta$  to form  $m$  samples from  $\Delta_i \times \Delta$ , and then calls  $M$ .  $M'$  is a p.p.t. distinguisher for  $\Delta_1, \Delta_2$ .  $\square$

## 2.2. Model of Secure Computation

We assume that there are  $n$  parties, each of which is a p.p.t. Turing machine (read-only input tape, read-only auxiliary tape, write-only output tape, random tape, one or more work tapes). The parties communicate by means of a broadcast channel; this can be modeled as an additional tape (communication tape) for each machine that is write-only for its owner and read-only for everyone else. When a party writes a message to this tape, we may say that the message has been “broadcast” or “posted.” A protocol begins with all  $n$  parties in their start states, alternates local computation with synchronous rounds of broadcast, and ends when all parties have reached their final states. The output of the protocol is the (common) value written on the output tapes of the processors.

We are concerned with the message complexity of a protocol. This is measured as the total number of bits written on the communication tapes during the execution of the protocol. Since our protocols are cryptographic, we state the message complexity in terms of the number of *encrypted bits* written on the communication tapes. For the protocols we consider, this is all or most of the communication that occurs, and it is also a convenient measure independent of advances in either encryption methods or cryptanalytic techniques. Alternatively, measures could be expressed using a security parameter: Given a security parameter  $k$ ,  $O(e)$  encrypted bits of communication is equivalent to  $O(ek)$  bits of communication for the protocols discussed in this paper.

When the parties  $[1 \dots n]$  begin with  $x_1, \dots, x_n$  on their input tapes,  $a_1, \dots, a_n$  on their auxiliary tapes,  $r_1, \dots, r_n$  on their random tapes, and execute protocol  $P$ , then we let  $\text{VIEW}^P[(x_1, a_1), \dots, (x_n, a_n) \mid r_1, \dots, r_n]$  denote the concatenation of the contents

of all input, auxiliary, and communication tapes at the start of the protocol, and after every round of broadcast. The view of a subset of parties  $S$ , denoted

$$\text{VIEW}_S^P[(x_1, a_1), \dots, (a_n, x_n) \mid r_1, \dots, r_n]$$

is the restriction to those tapes readable by  $S$ . We let  $\text{VIEW}_S^P[(x_1, a_1), \dots, (x_n, a_n)]$  denote the distribution of views with respect to a uniformly random choice of random tapes  $r_1, \dots, r_n$ ;  $\text{VIEW}_S^P[(x_1, a_1), \dots, (x_n, a_n)]$  is defined similarly.

For notational convenience, we typically omit the reference to auxiliary tapes in the notation for views. For example, we write  $\text{VIEW}_S^P[x_1, \dots, x_n]$  to denote the distribution of views of  $S$  of executions of the protocol. In this paper the auxiliary tape of a party always holds cryptographic keys (i.e., the public key and its own private key).

We say that a protocol is “private” if its execution reveals no useful information to any proper subset of (polynomial bounded) gossiping processors. More precisely, suppose that protocol  $P$  computes a function  $f$ ; i.e., when each party  $i$  begins with  $u_i$  on its input tape, all parties end with  $f(u_1, \dots, u_n)$  on their output tapes. Let  $S$  be a proper subset of parties; without loss of generality  $S = [1 \dots t]$ ,  $t < n$ . Then  $P$  is private if, for all  $u_1, \dots, u_t$ , for all  $v_{t+1}, w_{t+1}, \dots, v_n, w_n$  such that  $f(u_1, \dots, u_t, v_{t+1}, \dots, v_n) = f(u_1, \dots, u_t, w_{t+1}, \dots, w_n)$ , the distributions of views

$$\text{VIEW}_S^P[u_1, \dots, u_t, v_{t+1}, \dots, v_n] \text{ and } \text{VIEW}_S^P[u_1, \dots, u_t, w_{t+1}, \dots, w_n]$$

are computationally indistinguishable. By this definition, the messages seen by any subset of gossipers during the protocol cannot help them guess the other's inputs (beyond what they already know from their inputs together with the final output).

### 3. Joint Encryption

A joint encryption scheme for  $n$  parties  $[1 \dots n]$  is given by a public key  $K_{\text{pub}}$ ,  $n$  private keys  $K_1, \dots, K_n$ , a collection of encryption functions  $\{E_S: S \subseteq [1 \dots n]\}$  and a collection of decryption functions  $\{D_i: i \in [1 \dots n]\}$ . The encryption functions are defined on bitstrings (i.e., have domain  $\{0, 1\}^*$ ), and are possibly probabilistic. The connection among these functions is given by

$$D_i(E_S(m)) = E_{S-\{i\}}(m)$$

for all  $m \in \{0, 1\}^*$  and for all  $i \in S$ . It should be easy to compute any  $E_S$  given only the public key  $K_{\text{pub}}$ , but hard to compute any  $D_i$  without the private key  $K_i$  (where hardness is with respect to a security parameter  $k$  that governed the choice of public and private keys). Lastly, it should always be easy to compute  $m$  from  $E_\emptyset(m)$ .

In this section we describe a joint encryption scheme with special algebraic properties that is the main tool used for our message-efficient secure computation protocol. This encryption scheme is related to ElGamal encryption [5], ElGamal encryption with a composite modulus [10], and encryption based on quadratic residuosity [9]. A group encryption scheme of Desmedt and Frankel [3] is based on ElGamal encryption, but lacks the additional properties that we need for our secure computation protocol (i.e., Claims 2 and 3 of Section 3.3).

### 3.1. Our Joint Encryption Scheme

Let  $k$  be a security parameter. Let  $N = pq$ ,  $p \equiv q \pmod{4}$ , where  $p$  and  $q$  are primes of length  $k$  such that  $\gcd(p-1, q-1) = O(1)$ . Let  $g \in Z_N^*$  have Jacobi symbol  $+1$  modulo  $N$ , and let  $g$  have order  $\lambda = \Theta(N)$  in  $Z_N^*$ , i.e.,  $\langle g \rangle = \{g^r \pmod{N} : r \in Z_N\}$  has size  $\lambda = \text{lcm}(p-1, q-1)/c$  for some constant  $c \geq 1$ .

The public key is  $[N, g^{x_1} \pmod{N}, \dots, g^{x_n} \pmod{N}]$ . The trapdoor information for  $D_i$  is  $x_i$ , which is known to party  $i$ . No party knows the factorization of  $N$ . Encryption of a zero is given by

$$E_S(0) = \left[ g^r \pmod{N}, g^{r'} \pmod{N}, s^2 \left( \prod_{j \in S} g^{x_j} \right)^r \pmod{N}, s \left( \prod_{j \in S} g^{x_j} \right)^{r'} \pmod{N} \right]$$

for  $r, r' \in_R Z_N, s \in_R Z_N^*$ . Encryption of a one is given by

$$E_S(1) = \left[ g^r \pmod{N}, g^{r'} \pmod{N}, -s^2 \left( \prod_{j \in S} g^{x_j} \right)^r \pmod{N}, s \left( \prod_{j \in S} g^{x_j} \right)^{r'} \pmod{N} \right]$$

for  $r, r' \in_R Z_N, s \in_R Z_N^*$ . Decryption is given by

$$D_i([\alpha, \beta, \gamma, \delta]) = [\alpha, \beta, \gamma \alpha^{-x_i} \pmod{N}, \delta \beta^{-x_i} \pmod{N}].$$

The third and fourth components of  $E_\emptyset(b)$  enable the value of  $b$  to be computed easily.

It is important that  $N$  be hard to factor, and that the Jacobi symbol modulo  $N$  of  $-1$  be  $+1$ ; otherwise, our scheme is provably insecure. The requirement that  $N = pq$ ,  $p \equiv q \pmod{4}$ , was intended to provide these properties while minimizing the length of  $N$ .

It is important that the factorization of  $N$  be unknown to all parties, because  $b$  is easy to determine from  $E_S(b)$  given  $p$  and  $q$  (see Section 3.4). The keys for this scheme could be computed and distributed by a central server that knows  $n$  and a security parameter  $k$ . The center could choose the factors of  $N$  of length  $k$ , choose  $g, x_1, \dots, x_n$  appropriately, compute and publish the public key, and securely distribute the private keys. After distributing the keys, the center could destroy all of its information, i.e., the factorization of  $N$  does not need to be stored anywhere for any future purpose.

### 3.2. Distributions of Jointly Encrypted Bits

In this section we establish a few facts about some probability distributions associated with our joint encryption scheme. Let  $N, p, q, g$  satisfy the requirements for our joint encryption scheme:  $g$  has order  $\lambda = \Theta(N)$  in  $Z_N^*$ ,  $N = pq$ ,  $p \equiv q \pmod{4}$ , where  $p$  and  $q$  are primes of length  $k$  such that  $\gcd(p-1, q-1) = O(1)$ . Let  $U[X]$  be the uniform distribution on the set  $X$ . For any  $z \in \langle g \rangle$ , define  $\Theta_z$  to be the following distribution on  $\langle g \rangle$ :  $[zg^r \pmod{N} : r \in_R Z_N]$ .

**Fact 3.** *The distribution of jointly encrypted bits for a given bit is  $\Theta_1 \times \Theta_1 \times U[Z_N^*]$ .*

**Proof.** The first term of a joint encryption is  $g^r \bmod N$  for  $r \in_R Z_N$ , i.e., an element with distribution  $\Theta_1$ . The second term is similar. The fourth term has distribution  $U[Z_N^*]$  since it multiplies a value in  $Z_N^*$  by  $s \in_R Z_N^*$ . The third term is deterministic given the other three.  $\square$

**Fact 4.** For all  $z \in (g)$ ,  $\Theta_z$  and  $U[(g)]$  are statistically indistinguishable.

**Proof.** Let  $\mu = ((p-1)(q-1))/\lambda$ . If  $zg^r \bmod N = y$ , then  $zg^{r'} \bmod N = y$  for every  $r \equiv r' \bmod \lambda$ . It follows that under  $\Theta_z$ ,  $\lambda - p - q + 1$  elements are selected with probability  $\mu/N$  each, and  $p + q - 1$  elements are selected with probability  $(\mu + 1)/N$  each. Under  $U$ , all  $\lambda$  elements are selected with probability  $1/\lambda$  each.

$$\begin{aligned} |\Theta_z - U[(g)]| &= \sum_{x \in (g)} |\Theta_z[x] - U[(g)][x]| \\ &= (\lambda - p - q + 1) \left| \frac{1}{\lambda} - \frac{\mu}{N} \right| + (p + q - 1) \left| \frac{\mu + 1}{N} - \frac{1}{\lambda} \right| \\ &= O(2^{-k}). \end{aligned} \quad \square$$

**Fact 5.** For all  $z_1, \dots, z_m \in (g)$ ,  $m = \text{poly}(k)$ ,  $\Theta_{z_1} \times \dots \times \Theta_{z_m}$  and  $U[(g)]^m$  are statistically indistinguishable.

**Proof.** Under  $\Theta_{z_1} \times \dots \times \Theta_{z_m}$ , at least  $(\lambda - p - q + 1)^m$  elements have probability at most  $(\mu/N)^m$ , and at most  $\lambda^m - (\lambda - p - q + 1)^m$  elements have probability at most  $((\mu + 1)/N)^m$ . Thus

$$\begin{aligned} |\Theta_{z_1} \times \dots \times \Theta_{z_m} - U[(g)]^m| &\leq (\lambda - p - q + 1)^m \left| \left( \frac{\mu}{N} \right)^m - \left( \frac{1}{\lambda} \right)^m \right| \\ &\quad + (\lambda^m - (\lambda - p - q + 1)^m) \left| \left( \frac{\mu + 1}{N} \right)^m - \left( \frac{1}{\lambda} \right)^m \right| \\ &= O(2^{-k}). \end{aligned} \quad \square$$

### 3.3. Properties of Our Joint Encryption Scheme

We identify some useful properties of our joint encryption scheme.

**Claim 1 (Compact).** The size of each encrypted bit is four elements of  $Z_N$ , independent of the number of participants.

**Claim 2 (Xor-Homomorphic).** From  $E_S(b)$  and  $E_S(b')$  it is easy to compute an element in  $\{z: D_S(z) = b \oplus b'\}$ .

**Proof.** If  $E_S(b) = [\alpha, \beta, \gamma, \delta]$  and  $E_S(b') = [\alpha', \beta', \gamma', \delta']$ , then

$$[\alpha\alpha' \bmod N, \beta\beta' \bmod N, \gamma\gamma' \bmod N, \delta\delta' \bmod N]$$

is an encryption of  $b \oplus b'$  under the keys of  $S$ .  $\square$

**Claim 3** (Blindable). *From  $E_S(b)$  it is easy to sample from  $\{z: D_S(z) = b\}$  with a distribution that is statistically indistinguishable from uniform.*

**Proof.** If  $[\alpha, \beta, \gamma, \delta]$  is a joint encryption using  $E_S$ , and  $r, r' \in_R Z_N, s \in_R Z_N^*$ , then

$$[\alpha', \beta', \gamma', \delta'] = \left[ \alpha g^r \bmod N, \beta g^{r'} \bmod N, \gamma s^2 \left( \prod_{j \in S} g^{x_j} \right)^r \bmod N, \delta s \left( \prod_{j \in S} g^{x_j} \right)^{r'} \bmod N \right]$$

is a joint encryption of the same value. The value  $\alpha'$  is drawn from  $\Theta_\alpha$ ,  $\beta'$  is drawn from  $\Theta_\beta$ ,  $\delta'$  is drawn from  $U[Z_N^*]$ , and  $\gamma'$  is deterministic given the other three and the value of the encrypted bit. From the preceding section, we know that  $\Theta_\alpha \times \Theta_\beta \times U[Z_N^*]$  is statistically indistinguishable from the uniform distribution  $U[(g)] \times U[(g)] \times U[Z_N^*]$  (and statistically indistinguishable from  $\Theta_1 \times \Theta_1 \times U[Z_N^*]$ , the distribution of joint encryptions).  $\square$

**Claim 4** (Witnessable). *There is a function  $W_i$  such that  $D_i(E_S(b))$  is easy to compute from  $E_S(b)$  and  $W_i(E_S(b))$ . Furthermore, each witness is half the size of an encryption (i.e., two elements of  $Z_N$ ).*

**Proof.** If  $E_S(b) = [\alpha, \beta, \gamma, \delta]$ , then  $D_i(E_S(b))$  can be easily computed from

$$[\alpha^{-x_i} \bmod N, \beta^{-x_i} \bmod N]$$

for any  $i \in S$ .  $\square$

### 3.4. Security of Our Joint Encryption Scheme

A probabilistic encryption scheme is GM-secure [9] if, for any two plaintexts, the distributions of ciphertexts are computationally indistinguishable (where the keys are chosen independently of the plaintexts). We say that a joint encryption scheme is GM-secure if  $E_S$  is GM-secure for every nonempty  $S \subseteq [1 \dots n]$ .

The security of our joint encryption scheme can be related to the security of ElGamal encryption with a composite modulus. We define ElGamal encryption with a composite modulus as follows:  $E(M) = [g^r \bmod N, M g^{rx} \bmod N]$  for  $g, N$  chosen as in our joint encryption scheme, public key  $g^x \bmod N, r \in_R Z_N$ , and for  $M \in \mathcal{M}$ , where the message space  $\mathcal{M}$  is the set of elements of  $Z_N^*$  with Jacobi symbol  $+1$ .

**Claim 5** (Security). *If ElGamal encryption with a composite modulus is GM-secure, then our joint encryption scheme is GM-secure.*

**Proof.** Suppose, for purposes of establishing a contradiction, that ElGamal encryption with a composite modulus is GM-secure while our joint encryption scheme is not GM-secure. Then it would be easy to distinguish between composite ElGamal encryptions of  $+1$  and  $-1$ , since these can be easily converted into almost uniformly random joint encryptions of one and zero, as follows.



Let  $(g^r \bmod N, (-1)^b g^{rx} \bmod N)$  be a composite ElGamal encryption of  $(-1)^b$  (using ElGamal public key  $g^x \bmod N$  and  $r \in_R Z_N$ ). Then

$$[g^r \bmod N, g^{r'} \bmod N, s^2(-1)^b g^{rx} \bmod N, sg^{r'x} \bmod N]$$

is a joint encryption  $E_S(b)$  drawn from the proper distribution  $\Theta_1 \times \Theta_1 \times U[Z_N^*]$ , when  $r' \in_R Z_N, s \in_R Z_N^*$  (e.g., using joint public key

$$\left[ N, r_1, \dots, r_{|S|-1}, \left( g^x \prod_{i < |S|} r_i^{-1} \bmod N \right) \right]$$

for  $r_1, \dots, r_{|S|-1} \in_R (g)$ . □

However, when  $p \equiv q \equiv 3 \bmod 4$ , our encryption scheme (and composite ElGamal) is not GM-secure if composite quadratic character (residue versus nonresidue) is easy to compute. The attacker sees  $g^x \bmod N, \alpha = g^r \bmod N, \beta = g^{r'} \bmod N, \gamma = (-1)^b s^2 g^{rx} \bmod N, \delta = sg^{r'x} \bmod N$ , where  $b$  is the value of the encrypted bit (and where  $x = \sum_{i \in S} x_i$ ). Let  $QR_N(v) = 0$  if  $v$  is a quadratic residue modulo  $N$  and 1 otherwise. If  $QR_N(\cdot)$  is easy to compute, then the attacker can determine  $b = (QR_N(\alpha) * QR_N(g^x \bmod N)) \oplus QR_N(\gamma)$ .

The security of the original ElGamal public-key encryption scheme reduces to the difficulty of breaking an instance of the Diffie–Hellman key exchange scheme [4] (i.e., a problem that is no more difficult than but not known to be equivalent to the discrete log problem). McCurley [10] showed that ElGamal encryption with a suitably restricted composite modulus is secure against an adversary who could break the Diffie–Hellman key exchange, or could factor the modulus, but not both (see also [12]). However, this was a proof of security in the sense that no polynomial-time algorithm can invert a nonnegligible fraction of ciphertexts, and not GM-security (computational indistinguishability of ciphertexts). In fact, without restricting the message space to have Jacobi symbol  $+1$ , McCurley’s version of composite ElGamal is not GM-secure: the Jacobi symbol of an encrypted message could be computed.

### 3.5. Comments About Our Joint Encryption Scheme

An important property of other group-oriented encryption schemes is “threshold decryption,” i.e., encryption such that any sufficiently large subset of parties can decrypt. Our joint encryption scheme, as described above, does not have this property, i.e., the threshold for decryption is always  $n$ . However, if  $g$  and  $N$  are chosen as suggested by McCurley [10], then the technical condition for incorporating threshold decryption into our scheme is met. Specifically, McCurley uses  $g = 16, N = pq, p = 8r + 3, q = 8s - 1$  (where  $r, s$  have special structure), and this meets the condition of Desmedt and Frankel [3] for their modified shadow generation scheme based on Lagrange interpolation (i.e.,  $g$  has odd order in  $Z_N^*$ ). For our main result, message-efficient secure computation, we do not need the threshold decryption property.

Our joint encryption scheme uses four elements of  $Z_N$  to encrypt a single bit. Here is a possible two-element scheme:

$$E_S(b) = \left[ g^r \bmod N, (-1)^b \left( \prod_{j \in S} g^{x_j} \right)^r \bmod N \right]$$

for  $r \in_R Z_N$ . This scheme has the same properties of compactness, xor-homomorphism, blindness, and witnessability as the four-element scheme; it could be substituted into the message-efficient secure computation protocol in Section 4. Like the four-element scheme, it is GM-secure if composite ElGamal is GM-secure. If the four-element scheme is not GM-secure, then neither is the two-element scheme; we do not know whether the converse is true.

### 3.6. Notation

Since all joint encryptions in the remainder of this paper are with respect to the keys of all parties  $[1 \dots n]$ , we simplify the notation and write  $E(b)$  instead of  $E_{[1 \dots n]}(b)$ . We also often omit “mod  $N$ ” when obvious. Since our encryption scheme is probabilistic,  $E_S(0)$  and  $E_S(1)$  refer to almost uniformly random encryptions of these values (i.e., drawn from  $\Theta_1 \times \Theta_1 \times U[Z_N^*]$ ). Given an encryption  $E_S(b) = [\alpha, \beta, \gamma, \delta]$ , an almost uniformly random (“blinded”) encryption of the same value is denoted  $E_S(b \oplus 0)$  (i.e., drawn from  $\Theta_\alpha \times \Theta_\beta \times U[Z_N^*]$ ). We write  $g^x = \prod_{i=1}^n g^{x_i} \bmod N$ . We use “ $\sum$ ” to denote XOR, e.g.,  $\sum_{j=1}^n b_j = b_1 \oplus \dots \oplus b_n$ .

## 4. Message-Efficient Secure Computation

We now state our main result about secure computation with low message complexity in the cryptographic broadcast-only model. A boolean circuit consists of 1-ary NOT gates and 2-ary AND gates.

**Theorem 1.** *If ElGamal encryption with a composite modulus is GM-secure, then any boolean circuit with  $C$  gates can be privately evaluated by  $n$  parties using  $O(nC)$  encrypted bits of communication.*

We prove this theorem by presenting a circuit evaluation protocol and proving that it has the desired properties. The protocol is given in Section 4.1, and three lemmas in Section 4.2 prove correctness, communication efficiency, and privacy.

### 4.1. Protocol for Circuit Evaluation

The protocol begins with each party having its own input on its input tape, and its own private key  $x_i$  together with the public key  $[N, g^{x_1}, \dots, g^{x_n}]$  on its auxiliary tape. To start the protocol, each party broadcasts a joint encryption of its input bits. We show the encrypted output of any gate can be computed in a constant number of rounds from its encrypted inputs. For a NOT gate, the output can be found without any communication by XORing the encrypted input with a default encryption of a one (e.g.,  $[1, 1, -1, 1]$ ).

For an AND gate, suppose the encrypted gate inputs are  $\hat{u} = E(u)$  and  $\hat{v} = E(v)$ . The protocol ends with every party able to compute  $\hat{w} = E(u \wedge v)$ , as follows:

1. Each party  $i$  broadcasts  $\hat{b}_i = E(b_i)$  and  $\hat{c}_i = E(c_i)$ , where  $b_i, c_i \in_R \{0, 1\}$ .
2. Each party  $i$  broadcasts decryption witnesses  $W_i(\hat{u}')$  and  $W_i(\hat{v}')$ , where:
  - (a)  $\hat{u}' = E(u \oplus b_1 \oplus \cdots \oplus b_n)$ .
  - (b)  $\hat{v}' = E(v \oplus c_1 \oplus \cdots \oplus c_n)$ .
3. Each party  $i$  broadcasts  $\hat{w}_i = E(w_i) = E(0 \oplus (b_i \wedge c_1) \oplus \cdots \oplus (b_i \wedge c_n) \oplus (b_i \wedge v) \oplus (u \wedge c_i))$ , where:
  - (a)  $E(b_i \wedge c_j) = E(0)$  whenever  $b_i = 0$ .
  - (b)  $E(b_i \wedge c_j) = E(c_j)$  whenever  $b_i = 1$ .
  - (c)  $E(b_i \wedge v)$  and  $E(u \wedge c_i)$  are computed similarly.
4. Each party  $i$  can now compute the encrypted output  $\hat{w}$  of the AND gate:
  - (a)  $w' = (u \oplus b_1 \oplus \cdots \oplus b_n) \wedge (v \oplus c_1 \oplus \cdots \oplus c_n) = u' \wedge v'$  (computable from step 2).
  - (b)  $\hat{w} = E(w' \oplus w_1 \oplus \cdots \oplus w_n)$ .

When the last gate in the circuit has been computed, all parties know a joint encryption of the circuit output. At this point, the parties broadcast decryption witnesses to enable all of them to compute the actual circuit output.

#### 4.2. Proof of Theorem 1

**Lemma 1.** *The protocol in Section 4.1 is correct.*

**Proof.** Correctness follows from the xor-homomorphic and witnessable properties of our joint encryption scheme, together with the distributivity of AND over XOR:  $w' = (u \oplus b_1 \oplus \cdots \oplus b_n) \wedge (v \oplus c_1 \oplus \cdots \oplus c_n) = (u \wedge v) \oplus \sum_i (u \wedge c_i) \oplus \sum_i (b_i \wedge v) \oplus \sum_{i,j} (b_i \wedge c_j) = (u \wedge v) \oplus \sum_i w_i$ , where  $w_i = (b_i \wedge v) \oplus (u \wedge c_i) \oplus \sum_j (b_i \wedge c_j)$ .  $\square$

**Lemma 2.** *The protocol in Section 4.1 satisfies the communication claim of Theorem 1.*

**Proof.** The proof follows from the compactness of our joint encryption scheme. No communication is needed for each NOT gate. Each AND gate requires two rounds of communication (since the broadcasts for steps 2 and 3 can be performed in parallel), and message complexity  $4n$  encrypted bits (actually, three encryptions and two decryption witnesses per party, where each witness is half the length of an encryption).  $\square$

It remains to be shown that the protocol in Section 4.1 is private. Intuitively, privacy is achieved because everything remains encrypted throughout the protocol except for certain values that are completely random. Nothing is decrypted (or even sent) when a NOT gate is computed. When an AND gate is computed, the only values that are decrypted are  $u'$  and  $v'$ . These were derived from the original inputs to the gate by “masking” them with random  $b_i, c_i$  values chosen by each party. Fully decrypting these masked inputs—while critical to the efficient evaluation of the gate in our protocol—gives no information about the real inputs. Other than these random masked inputs, no values are decrypted during the computation of an AND gate.

In the proof of Lemma 3, we say that each input line is “owned” by one of the  $n$  parties, i.e., the party that knows the boolean value to be supplied on that line.

**Lemma 3.** *IF ElGamal encryption with a composite modulus is GM-secure, then the protocol in Section 4.1 is private.*

**Proof.** We show a direct reduction from the distinguishability of a single encrypted bit. Suppose that a circuit  $C$  with  $n'$  inputs exists for which distinguishable views are possible. That is, values  $\gamma_1, \dots, \gamma_{i^*}, \delta_{i^*+1}, \dots, \delta_{n'}, \varepsilon_{i^*+1}, \dots, \varepsilon_{n'}$  exist such that

$$C(\gamma_1, \dots, \gamma_{i^*}, \delta_{i^*+1}, \dots, \delta_{n'}) = C(\gamma_1, \dots, \gamma_{i^*}, \varepsilon_{i^*+1}, \dots, \varepsilon_{n'})$$

while  $\text{VIEW}_{[1 \dots t]}^P[\gamma_1, \dots, \gamma_{i^*}, \delta_{i^*+1}, \dots, \delta_{n'}]$  and  $\text{VIEW}_{[1 \dots t]}^P[\gamma_1, \dots, \gamma_{i^*}, \varepsilon_{i^*+1}, \dots, \varepsilon_{n'}]$  are computationally distinguishable; here the ownership of input lines to  $C$  has been chosen so that input lines  $[1 \dots i^*]$  are owned by parties in  $[1 \dots t]$  and input lines  $[i^* + 1 \dots n']$  are owned by parties in  $[t + 1 \dots n]$ . Let  $\hat{b} = E(b)$  be the joint encryption of an unknown bit. We show how to sample efficiently and almost uniformly from a distribution which is  $\text{VIEW}_{[1 \dots t]}^P[\gamma_1, \dots, \gamma_{i^*}, \delta_{i^*+1}, \dots, \delta_{n'}]$  whenever  $b = 0$ , and which is  $\text{VIEW}_{[1 \dots t]}^P[\gamma_1, \dots, \gamma_{i^*}, \varepsilon_{i^*+1}, \dots, \varepsilon_{n'}]$  whenever  $b = 1$ . Computational distinguishability of the two distributions of views implies computational distinguishability of jointly encrypted bits; this contradicts the assumed GM-security of ElGamal with a composite modulus (by Claim 5 of the preceding section).

Although we have access to the private keys of parties  $[1 \dots t]$ , our simulation in fact requires no private keys at all. We simulate the contents of the input and communication tapes of an execution of the protocol as follows. Initially, the simulated broadcast of each party  $i$ ,  $1 \leq i \leq n$ , is  $\hat{d}_j$  for every input line  $j$  owned by party  $i$ , where:

- $\hat{d}_j = E(\gamma_j)$  for all  $j$ ,  $1 \leq j \leq i^*$ .
- $\hat{d}_j = E(\delta_j)$  for all  $j$ ,  $i^* + 1 \leq j \leq n'$ , such that  $\delta_j = \varepsilon_j$ .
- $\hat{d}_j = E(b \oplus 0)$  for all  $j$ ,  $i^* + 1 \leq j \leq n'$ , such that  $\delta_j = 0$  and  $\varepsilon_j = 1$ .
- $\hat{d}_j = E(b \oplus 1)$  for all  $j$ ,  $i^* + 1 \leq j \leq n'$ , such that  $\delta_j = 1$  and  $\varepsilon_j = 0$ .

Notice that each encrypted input has a known value from among  $\{0, 1, b, 1 - b\}$ . This will also be true of the output of each gate, a fact that we exploit in our simulation of each AND gate.

There is no communication for a NOT gate, so the simulation does nothing. Notice that it is easy to determine the value of the encrypted output of a NOT gate from among  $\{0, 1, b, 1 - b\}$  if the value of the encrypted input to the gate is known from among this set:

NOT input	NOT output
$E(0)$	$E(1)$
$E(1)$	$E(0)$
$E(b)$	$E(1 - b)$
$E(1 - b)$	$E(b)$

For computing each AND gate of  $C$ , suppose that the jointly encrypted inputs are  $\hat{u} = E(u) = [g^{r_u}, g^{r'_u}, (-1)^u s_u^2 g^{x r_u}, s_u g^{x r'_u}]$  and  $\hat{v} = E(v) = [g^{r_v}, g^{r'_v}, (-1)^v s_v^2 g^{x r_v}, s_v g^{x r'_v}]$ . Further suppose that  $u$  and  $v$  have known values from among  $\{0, 1, b, 1 - b\}$ . The communication tape contents of all  $n$  parties for the AND gate protocol can be computed as follows:

1. The first simulated broadcast for each party  $i$  is  $\hat{b}_i, \hat{c}_i$ , where:
  - (a)  $b_i, c_i \in_R \{0, 1\}$  for all  $i, 1 \leq i \leq n$ .
  - (b)  $r_{b_i}, r'_{b_i}, r_{c_i}, r'_{c_i} \in_R \mathbb{Z}_N; s_{b_i}, s_{c_i} \in_R \mathbb{Z}_N^*$  for all  $i, 1 \leq i \leq n$ .
  - (c) For all  $i, 1 \leq i \leq n - 1, \hat{b}_i = E(b_i) = [g^{r_{b_i}}, g^{r'_{b_i}}, (-1)^{b_i} s_{b_i}^2 g^{x r_{b_i}}, s_{b_i} g^{x r'_{b_i}}]$ .
  - (d) For all  $i, 1 \leq i \leq n - 1, \hat{c}_i = E(c_i) = [g^{r_{c_i}}, g^{r'_{c_i}}, (-1)^{c_i} s_{c_i}^2 g^{x r_{c_i}}, s_{c_i} g^{x r'_{c_i}}]$ .
  - (e)  $\hat{b}_n = E(b_n \oplus u) = [g^{r_{b_n} g^{r_u}}, g^{r'_{b_n} g^{r'_u}}, (-1)^{b_n} s_{b_n}^2 g^{x r_{b_n}} (-1)^u s_u^2 g^{x r_u}, s_{b_n} g^{x r'_{b_n}} s_u g^{x r'_u}]$ .
  - (f)  $\hat{c}_n = E(c_n \oplus v) = [g^{r_{c_n} g^{r_v}}, g^{r'_{c_n} g^{r'_v}}, (-1)^{c_n} s_{c_n}^2 g^{x r_{c_n}} (-1)^v s_v^2 g^{x r_v}, s_{c_n} g^{x r'_{c_n}} s_v g^{x r'_v}]$ .
2. The second simulated broadcast for each party  $i$  is  $W_i(\hat{u}'), W_i(\hat{v}')$ , where:
  - (a)  $W_i(\hat{u}') = [(g^{x_i})^{-(r_{b_1} + \dots + r_{b_n})}, (g^{x_i})^{-(r'_{b_1} + \dots + r'_{b_n})}]$ .
  - (b)  $W_i(\hat{v}') = [(g^{x_i})^{-(r_{c_1} + \dots + r_{c_n})}, (g^{x_i})^{-(r'_{c_1} + \dots + r'_{c_n})}]$ .
3. The third simulated broadcast for each party  $i$  is  $\hat{w}_i = E(w_i \oplus 0)$  where:
  - (a) For  $1 \leq i \leq n - 1, w_i = (u \wedge c_i) \oplus (b_i \wedge v) \oplus (b_i \wedge c_1) \oplus \dots \oplus (b_i \wedge c_{n-1}) \oplus (b_i \wedge (c_n \oplus v)) = (u \wedge c_i) \oplus (b_i \wedge c_1) \oplus \dots \oplus (b_i \wedge c_n)$ .
  - (b)  $w_n = (u \wedge (c_n \oplus v)) \oplus ((b_n \oplus u) \wedge v) \oplus ((b_n \oplus u) \wedge c_1) \oplus \dots \oplus ((b_n \oplus u) \wedge c_{n-1}) \oplus ((b_n \oplus u) \wedge (c_n \oplus v)) = (u \wedge v) \oplus (b_n \wedge c_1) \oplus \dots \oplus (b_n \wedge c_n) \oplus (u \wedge c_1) \oplus \dots \oplus (u \wedge c_{n-1})$ .
  - (c)  $E(b_i \wedge c_j) = E(0)$  whenever  $b_i = 0$ .
  - (d)  $E(b_i \wedge c_j) = E(c_j)$  whenever  $b_i = 1$ .
  - (e)  $E(b_i \wedge v)$  and  $E(u \wedge c_i)$  are computed similarly.
  - (f)  $E(u \wedge v)$  is computed from the known values of  $u$  and  $v$ , according to the following table:

$\wedge$	$E(0)$	$E(1)$	$E(b)$	$E(1 - b)$
$E(0)$	$E(0)$	$E(0)$	$E(0)$	$E(0)$
$E(1)$	$E(0)$	$E(1)$	$E(b)$	$E(1 - b)$
$E(b)$	$E(0)$	$E(b)$	$E(b)$	$E(0)$
$E(1 - b)$	$E(0)$	$E(1 - b)$	$E(0)$	$E(1 - b)$

4. The encrypted output of the gate is computed to be  $\hat{w}$ , where:
  - (a)  $w' = (b_1 \oplus \dots \oplus b_n) \wedge (c_1 \oplus \dots \oplus c_n)$ .
  - (b)  $\hat{w} = E(w' \oplus w_1 \oplus \dots \oplus w_n)$ .

These four steps give the contents of all communication tapes for an almost uniformly random execution of the AND gate protocol. Furthermore, the table in step 3(f) gives the value of the encrypted output of the gate from among  $\{0, 1, b, 1 - b\}$ , as needed for later AND gates in the circuit.

Together with the auxiliary tapes of parties  $[1 \dots t]$ , this allows us to sample efficiently and almost uniformly from a distribution which is  $\text{VIEW}_{[1 \dots t]}^P[\gamma_1, \dots, \gamma_i, \delta_{i^*+1}, \dots, \delta_{n'}]$  whenever  $b = 0$ , and which is  $\text{VIEW}_{[1 \dots t]}^P[\gamma_1, \dots, \gamma_i, \varepsilon_{i^*+1}, \dots, \varepsilon_{n'}]$  whenever  $b = 1$ . In

fact, the distribution of real views is statistically indistinguishable from the distribution of simulated views, as the following argument shows. First, note that the entire transcript of the protocol, real or simulated, is given by the initial broadcasts and the broadcasts for every AND subprotocol. Second, note that for every encrypted bit broadcast either initially or during an AND subprotocol, the value of the bit is the same for the real and simulated views. The only difference between real and simulated views is that encryptions are selected from slightly different distributions.

For each AND gate, the table below gives the distributions from which the real and simulated views are drawn for each party in each round, where  $z, z', y_1, \dots, y_n, y'_1, \dots, y'_n$  are specific values in  $(g)$ . For the purposes of the proof, the actual values of  $z, z', y_1, \dots, y_n, y'_1, \dots, y'_n$ , or any dependencies among them, are irrelevant:

Round	Real view $1 \leq i \leq n$	Simulated view $1 \leq i \leq n-1$	Simulated view $n$
[1]	$\Theta_1 \times \Theta_1 \times U[Z_N^*]$	$\Theta_1 \times \Theta_1 \times U[Z_N^*]$	$\Theta_z \times \Theta_{z'} \times U[Z_N^*]$
[2]	Deterministic	Deterministic	Deterministic
[3]	$\Theta_{y_i} \times \Theta_{y'_i} \times U[Z_N^*]$	$\Theta_{z_i} \times \Theta_{z'_i} \times U[Z_N^*]$	$\Theta_{z_n} \times \Theta_{z'_n} \times U[Z_N^*]$

Each encrypted input bit  $\hat{d}_j$  from the initial broadcast is drawn from the real view from  $\Theta_1 \times \Theta_1 \times U[Z_N^*]$ . For the simulated view, it is drawn from  $\Theta_\alpha \times \Theta_\beta \times U[Z_N^*]$  whenever  $i^* < j \leq n'$  and  $\delta_j \neq \varepsilon_j$ , where  $[\alpha, \beta, \gamma, \delta]$  is the encryption of the unknown bit  $b$ . Otherwise, the encrypted input bit  $\hat{d}_j$  is drawn for the simulated view from  $\Theta_1 \times \Theta_1 \times U[Z_N^*]$ .

Using Facts 1, 2, and 5 established previously, the distributions of real and simulated views are statistically indistinguishable. A p.p.t. distinguisher for the two distributions of real views would thus give a p.p.t. distinguisher for the two distributions of simulated views. This would give a p.p.t. algorithm to distinguish the value of the encrypted bit  $b$ . This implies the GM-insecurity of our joint encryption scheme, contradicting the assumed GM-security of composite ElGamal.  $\square$

## 5. Concluding Remarks

The same decrease in message complexity is possible using our joint encryption scheme with a circuit-scrambling protocol. Specifically, our joint encryption can serve as the bit commitment scheme underlying the secure computation protocol of Chaum *et al.* [1]. Only a single commitment accompanies each truth-table row as it passes from party to party, representing the XOR of modifications performed by all parties thus far. The size of the scrambled circuit remains  $O(C)$  after each scramble, for a total message complexity of  $O(nC)$  encrypted bits. Although the message complexities are the same, the round complexities for the two approaches are incomparable:  $O(n)$  rounds for circuit scrambling, and  $O(D)$  for gate-by-gate, where  $D$  is the depth of the circuit being computed.

In this paper we have assumed a “writer” measure of message complexity. If one party

posts an encrypted bit to a publicly readable bulletin board, then the protocol is charged one encrypted bit. The same charge applies no matter how many of the other parties ever read that posted bit. It is reasonable to consider an alternative “reader” measure of message complexity, in which the protocol is charged  $m$  encrypted bits if a single encrypted bit is read by  $m$  of the other parties. With respect to the reader measure, the linear gain for Theorem 1 disappears; posting messages that are read by all other parties seems to be an essential feature of this approach. However, a linear gain still holds using our joint encryption scheme in the circuit-scrambling protocol of Chaum *et al.* This is because most broadcasts are read by only one other party, i.e., to pass a scrambled circuit from one party to the next.

An interesting open question is whether a reduction in message complexity for secure computation is possible under weaker cryptographic assumptions. It would also be interesting to understand message complexity requirements better for secure computation in the cryptographic setting versus stronger adversaries.

### Acknowledgments

We thank Ronald Cramer for suggesting the alternative encryption scheme described in Section 3.5, and for other helpful comments and discussions. Thanks also to Berry Schoenmakers, Moti Yung, and the anonymous referees for useful comments.

### References

- [1] D. Chaum, I. Damgård, and J. van de Graaf, Multiparty computations ensuring privacy of each party's input and correctness of the result, *Advances in Cryptology—CRYPTO '87 Proceedings* (Lecture Notes in Computer Science, Vol. 293), ed. C. Pomerance, pp. 87–119, Springer-Verlag, Berlin, 1988.
- [2] Y. Desmedt, Society and group oriented cryptography: a new concept, *Advances in Cryptology—CRYPTO '87 Proceedings* (Lecture Notes in Computer Science, Vol. 293), ed. C. Pomerance, pp. 120–127, Springer-Verlag, Berlin, 1988.
- [3] Y. Desmedt and Y. Frankel, Threshold cryptosystems, *Advances in Cryptology—CRYPTO '89 Proceedings* (Lecture Notes in Computer Science, Vol. 435), ed. G. Brassard, pp. 307–315, Springer-Verlag, Berlin, 1990.
- [4] W. Diffie and M. Hellman, New directions in cryptography, *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.
- [5] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Transactions on Information Theory*, 31:469–472, 1985.
- [6] Z. Galil, S. Haber, and M. Yung, Cryptographic computation: secure fault-tolerant protocols and the public-key model, *Advances in Cryptology—CRYPTO '87 Proceedings* (Lecture Notes in Computer Science, Vol. 293), ed. C. Pomerance, pp. 135–155, Springer-Verlag, Berlin, 1988.
- [7] O. Goldreich, S. Micali, and A. Wigderson, How to play any mental game, *Proceedings of the 19th Annual Symposium on Theory of Computing*, 1987, pp. 218–229.
- [8] O. Goldreich and R. Vainish, How to solve any protocol problem—an efficiency improvement, *Advances in Cryptology—CRYPTO '87 Proceedings* (Lecture Notes in Computer Science, Vol. 293), ed. C. Pomerance, pp. 73–86, Springer-Verlag, Berlin, 1988.
- [9] S. Goldwasser and S. Micali, Probabilistic encryption, *Journal of Computer and System Sciences*, 28:270–299, 1984.
- [10] K. McCurley, A key distribution system equivalent to factoring, *Journal of Cryptology*, 1:95–105, 1988.

- [11] S. Micali, Fair public-key cryptosystems, *Advances in Cryptology—CRYPTO '92 Proceedings* (Lecture Notes in Computer Science, Vol. 740), ed. E. Brickell, pp. 114–139, Springer-Verlag, Berlin, 1993.
- [12] Z. Shmueli, Composite Diffie–Hellman public-key generating systems are hard to break, Technical Report #356, Technion—Israel Institute of Technology, February 1985.
- [13] A. Yao, Protocols for secure computations, *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, 1982, pp. 160–164.
- [14] A. Yao, How to generate and exchange secrets, *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, 1986, pp. 162–167.