Controlled Gradual Disclosure Schemes for Random Bits and Their Applications

*Richard Cleve**
International Computer Science Institute
1947 Center St., Suite 600
Berkeley, CA 94704-1105, U.S.A.

Abstract

We construct a protocol that enables a secret bit to be revealed gradually in a very controlled manner. In particular, if Alice possesses a bit S that was generated randomly according to the uniform distribution and $\frac{1}{2} < p_1 < \cdots < p_m = 1$ then, using our protocol with Bob, Alice can achieve the following. The protocol consists of m stages and, after the i-th stage, Bob's best prediction of S, based on all his interactions with Alice, is correct with probability exactly p_i (and a reasonable condition is satisfied in the case where S is not initially uniform). Furthermore, under an intractability assumption, our protocol can be made "oblivious" to Alice and "secure" against an Alice or Bob that might try to cheat in various ways. Previously proposed gradual disclosure schemes for single bits release information in a less controlled manner: the probabilities that represent Bob's confidence of his knowledge of S follow a random walk that eventually drifts towards 1, rather than a predetermined sequence of values.

Using controlled gradual disclosure schemes, we show how to construct an improved version of the protocol proposed by Luby, Micali and Rackoff for two-party secret bit exchanging ("How to Simultaneously Exchange a Secret Bit by Flipping a Symmetrically-Biased Coin", *Proc. 22nd Ann. IEEE Symp. on Foundations of Computer Science*, 1983, pp. 11–21) that is secure against additional kinds of attacks that the previous protocol is not secure against. Also, our protocol is more efficient in the number of rounds that it requires to attain a given level of security, and is proven to be asymptotically optimal in this respect.

*Research partially conducted while the author was at the University of Toronto, partially supported by an NSERC postgraduate scholarship.

G. Brassard (Ed.): Advances in Cryptology - CRYPTO '89, LNCS 435, pp. 573-588, 1990.

We also show how to use controlled gradual disclosure schemes to improve existing protocols for other cryptographic problems, such as multi-party function evaluation.

1 Introduction

Suppose that $S \in \{0,1\}$ and Alice knows the value of S, but Bob has no idea about what the value of S is in the sense that Bob's best guess of S is correct with probability $\frac{1}{2}$ (such a state of knowledge could be obtained by having Alice flip a fair coin and look at the outcome but not show it to Bob). We are concerned with multistage protocols, called *disclosure schemes*, that, informally, enable Alice to provide Bob with partial information about S at each stage. We say that, at a particular stage, Bob's *confidence* about the value of S is the probability that Bob's best guess of S is correct (initially, Bob's confidence is $\frac{1}{2}$). A *gradual* disclosure scheme is, informally, a disclosure scheme in which Bob's confidence changes in small increments from $\frac{1}{2}$ to 1.

An example of a gradual disclosure scheme (considered by Luby, Micali and Rackoff [10], and Vazirani and Vazirani [12]), which we shall refer to as the *biased coin scheme*, operates as follows. Alice constructs a coin C that is biased towards S by ε, so that each time it is flipped its value is S with probability $\frac{1}{2}+\varepsilon$. Then, at each stage of the protocol, Alice flips C and sends the outcome to Bob. At each stage, Bob's best estimate of S is the majority value of the outcomes of C that he has received so far, and Bob's confidence depends on the "strength" of this majority (i.e. the difference between the number of 1s and 0s). Bob's confidence does not necessarily increase as the stages progress, but the *expected* value of Bob's confidence does increase. Also, at each stage, Bob's confidence changes by at most ε (and, thus, "gradually" if ε is small). It can be shown that if $\varepsilon = \frac{1}{m}$ then, after $\omega(m^2 \log^2 m)$ stages, the expected value of Bob's confidence is very high: $1 - (\frac{1}{m})^{\omega(1)}$.

A gradual disclosure scheme is *oblivious* if Alice learns nothing about Bob's current best estimate of S or confidence that she could not determine independently of the particular execution of the protocol. The biased coin scheme described above is not oblivious since Alice can completely determine Bob's state of knowledge about S from the values of the coin flips. If a third party, Ted, is allowed to participate, the biased coin scheme can be made oblivious by having Ted flip the coin C and reveal the outcomes to Bob but not to Alice.

Under a reasonable intractability assumption, the biased coin scheme can be made modified to be oblivious without the presence of Ted. It can also be modified to be secure against an Alice or Bob that may try to cheat in several possible ways. Alice could attempt to not send the coin flips according to the appropriate distribution (for instance Alice might try to convince Bob that S is 0 when S is really 1). In order

to be able to deal with this possibility, it is assumed that Alice initially sends Bob a cryptographic commitment of S. It is then possible for Bob to determine if Alice is behaving "consistently" with respect to the value of this committed bit. Bob could attempt to learn more about S than he is supposed to at a particular stage. The above possibilities can all be prevented.

Informally, a *controlled* gradual disclosure scheme is a gradual disclosure scheme with the additional property that, at each stage of the protocol, Bob's confidence level is a predetermined value, independent of the particular execution of the protocol. That is, there is a sequence of probabilities $p_1, ..., p_m$ (which are parameters of the protocol) such that, after stage i, Bob's confidence is p_i. Note that the biased coin scheme described above is *not* controlled.

One obvious advantage of a controlled gradual disclosure scheme over the biased coin scheme is in the number or stages required to attain a given bound on the incremental changes in Bob's confidence level during the disclosure. By setting $p_i = \frac{1}{2} + \frac{i}{2m}$, the change in Bob's confidence between successive stages is bounded by $\frac{1}{2m}$ and the secret is revealed within m rounds, whereas, to attain this bound and reveal the secret with high probability with the biased coin scheme, $\omega(m^2 \log^2 m)$ stages are required. (In all protocols considered here, a stage consists of a constant number of rounds. Therefore, these bounds also translate into similar bounds in terms of rounds.)

Another important advantage of a controlled gradual disclosure scheme arises from the property that the sequence of probabilities representing Bob's confidence level follows predetermined values. In particular, we can use this property to construct an improved version of the protocol proposed by Luby, Micali and Rackoff [10] (also considered by Yao [15]) for two-party secret bit exchanging. More specifically, the improved secret bit exchanging protocol has (in addition to the desirable properties of the previous protocol) the following property. Even if one party, say Alice, obtains information (possibly from some events in the outside world) about what Bob's current knowledge of her secret is, then she still cannot infer more information about Bob's secret from this than otherwise. (Circumstances similar to this and their affect on protocols are considered by Halpern and Rabin [9].) The previous two-party secret bit exchanging protocol is very vulnerable to these circumstances, whereas our protocol is very secure against such circumstances.

In Section 2, we present more formal definitions about our assumptions and our model. In Section 3, we show how to construct a controlled gradual disclosure scheme that is oblivious as well secure against an Alice or Bob that might try to cheat. In Section 4, we informally explain how our controlled gradual disclosure scheme can be used to construct an improved version of the secret bit exchanging protocols proposed by Luby, Micali and Rackoff [10]. Also, we sketch the proof of a lower bound on the

number of rounds required to attain a particular level of security that implies that our protocol is asymptotically optimal in the number of rounds that it requires. In Section 5, we informally explain how our controlled gradual disclosure scheme can be used to improve previous protocols for multi-party function evaluation (considered by Yao [15], and Beaver and Goldwasser [1]).

2 Definitions

2.1 Protocols

We represent a *two-party protocol* as an interacting pair of Turing machines (A, B) with the following tapes and properties. A and B have individual (read-only) *input* tapes, (read/write) *work* tapes, (read-only) *random* tapes, and (write-only) *output* tapes. Also, there are two *communication* tapes, one which is write-only to A and read-only to B, and one which is write-only to B and read-only to A. The input tapes are initialized with the input to the protocol, and the random tapes are initialized with independent random sequences of bits; all other tapes are initialized with the null string.

Both A and B have *sleep* states as well as *final* states. When the protocol is executed, beginning with A, the parties take turns running, each one running until it enters its sleep state or final state and then the other one starts running. This process continues until both parties are in their final states.

We also require that the running time of the protocol be polynomial in the following sense. For any B' and $x, y \in \{0,1\}^*$, when (A, B') is executed with x and y on the respective input tapes, the total running time of A is bounded by a polynomial in $|x|$ and $|y|$. Also, a similar condition holds if A is replaced by A'.

Each turn of A running until entering a sleep state, followed by B running until entering a sleep state is called a *round*. For convenience, we may partition the rounds into *stages*, (which each consist of a number of consecutive rounds).

In the special case of a trusted third party (considered in Section 3.1), the scheme is an interacting *triple* of Turing machines (A, B, T), defined similarly as above, where (A, T) and (B, T) have private communication tapes.

2.2 Controlled Gradual Disclosure Schemes

A *controlled gradual disclosure scheme* is a protocol that has the following properties. The protocol is run with A's input tape initialized with a random bit S followed by a string of n 1s, and B's input tape also initialized by a string of n 1s. The protocol first runs for a *commitment* stage which consists of a constant number of rounds (with

respect to n). Informally, in this stage, Alice is committing her value of S to Bob. Alice cannot be prevented from choosing a different secret bit S' and revealing this to Bob. What a protocol *can* guarantee is that Alice behaves consistently relative to *some* fixed secret bit S' which she must determine during the commitment stage (otherwise, Bob detects that Alice is inconsistent). Following the commitment stage, are a series of stages numbered $1, ..., m$. On completion of the i-th stage, B outputs b_i. Loosely speaking, b_i represents Bob's knowledge about S (or S') after stage i. Alice may deviate from the protocol at any time (represented by replacing A by another Turing machine A') and, similarly, Bob may deviate from the protocol at any time.

We adopt the following terminology in order to simplify our presentation. We write $\delta(n) \preceq \gamma(n)$, if $\delta(n) \leq \gamma(n) + (\frac{1}{n})^{\omega(1)}$, and $\delta(n) \simeq \gamma(n)$ if $\delta(n) \preceq \gamma(n)$ and $\gamma(n) \preceq \delta(n)$.

There are four conditions that we require a controlled gradual scheme to satisfy:

Correctness: If A and B follow the protocol correctly then, for all i, $b_i \in \{0,1\}$ and $\Pr[b_i = S] \simeq p_i$.

Informally, this means that if Alice and Bob both follow the protocol correctly then, on completion of the i-th stage, Bob learns the equivalent of the outcome of one coin that is biased towards S with probability p_i.

Consistency of Secret: If B correctly follows the protocol then, for all i, either $b_i \in \{0,1\}$ and $\Pr[b_i = S'] \succeq p_i$, or $b_i = \text{CHEAT}$.

Informally, this means that, after the commitment stage, there is no strategy for Alice to modify the information that she discloses to Bob without this being detected by him.

Security of Secret (from B): If A correctly follows the protocol then, for all i, $\Pr[b_i = S] \preceq p_i$.

Informally, this means that there is no strategy for Bob that increases the amount of information that he learns about S.

Obliviousness of Disclosure (to A): If B correctly follows the protocol and $i_{\max} = \max\{i : b_i \neq \text{CHEAT}\}$ then $\Pr[b_{i_{\max}} = S'] \succeq p_{i_{\max}}$.

The significance of this condition is more subtle than the previous conditions. Suppose that, at some stage i of the protocol, $p_i = \frac{3}{4}$. Then, with probability $\frac{1}{4}$, Bob's best estimate of S at this stage is wrong. If Alice were to know that this has occurred then, by quitting the protocol at this stage, she would leave Bob having significant confidence in something that she knows is false. If the disclosure is oblivious then, whenever Alice quits at this stage, from her point of view, Bob's final estimate of her secret is correct with probability $\frac{3}{4}$.

The above definition of a controlled gradual disclosure scheme assumes that, before executing the protocol, from Bob's point of view, the prior distribution of S is uniformly random. The situation is more complicated if this is not the case. In particular, if Alice does not know what Bob's prior information about S is then no protocol can have the property that Bob's confidence level follows a predetermined sequence of values. On the other hand, any controlled gradual disclosure scheme in the above sense (i.e. that satisfies the above properties when S is uniformly random) will satisfy a reasonable property when it is executed with with S chosen according to an arbitrary distribution. Intuitively, the property is that the information that Bob learns about S from the protocol after stage i is equivalent to Bob learning the outcome of a single coin that is biased towards S with probability p_i. Although this information may combine with Bob's prior information about S in different possible ways, yielding different possible confidence levels for Bob's knowledge about S, the amount of *new* information that Bob obtains is, in a reasonable sense, the same. This property is best expressed in terms of *likelihoods*, where the likelihood of an event is $l(p) = \log(\frac{p}{1-p})$ where p is the probability of the event. Then, after stage i, Bob's likelihood that $b_i = S$ satisfies $l(\Pr[b_i = S|\text{Bob's prior information}]) \simeq l(p_i) + l(q)$, where q is Bob's prior probability that $S = b_i$.

2.3 Complexity Theoretic Assumptions

For concreteness, let us base our scheme on the difficulty of determining certain quadratic residues.

We assume the **Quadratic Residuosity Conjecture**, which is that there is no probabilistic polynomial-time (in n) algorithm that achieves the following. The input to the algorithm is n, $p \cdot q$, where p and q are randomly chosen n-bit primes (and p and q are not explicitly given to the algorithm), and x, a random element of $\mathbf{Z}'_{p \cdot q}$. The goal of the algorithm is to determine with probability $\frac{1}{2} + (\frac{1}{n})^{O(1)}$ whether or not x is a quadratic residue (i.e. whether $x = y^2$, for some $y \in \mathbf{Z}'_{p \cdot q}$).

3 A Controlled Gradual Disclosure Scheme for a Random Bit

In this section, we construct a controlled gradual disclosure scheme that is oblivious as well secure against an Alice or Bob that try to cheat. Intuitively, the main idea behind our protocol is to simulate the flips of a special coin that adjusts its bias each time it is flipped. The new bias of the coin depends on the outcome the previous time it was flipped.

Let $\frac{1}{2} < p_1 < \cdots < p_m = 1$. Our protocol operates in m stages and, after the i-th stage is completed, achieves the following conditions. Bob's best guess of S (based on the information that Bob has seen so far) is correct with probability p_i. Moreover, Alice's best guess of what Bob's best guess of S is (based on the information that Alice has seen so far) is correct with probability p_i.

In Section 3.1, we describe how to adjust the biases of the coin so as to obtain the desired behavior, and, to simplify the presentation of this, we make the assumption that a trusted third party is present. In Section 3.2, we describe how to implement the protocol without a trusted third party.

3.1 First Implementation (with Trusted Third Party)

Assume that there is an honest third party, Ted, trusted by both Alice and Bob (in Section 3.2, the protocol will be modified to work without the presence of Ted). Initially, Alice sends Ted a copy of S, and Ted sends Bob a sequence of bits $C_1, ..., C_m$, which can intuitively be viewed as outcomes of a coin whose bias "evolves" each time it is flipped. We would like the biases of the coin to be such that, for all $i \in \{1, ..., m\}$, after seeing the outcomes of $C_1, ...C_i$, no matter what they are, Bob's best guess of S is the outcome of C_i, and for this guess to be correct with probability exactly p_i. That is, for all $x_1...x_i \in \{0,1\}^i$ and $x \in \{0,1\}$,

$$\Pr[S = x | C_1...C_i = x_1...x_i] = \begin{cases} p_i & \text{if } x = x_i \\ 1 - p_i & \text{if } x \neq x_i. \end{cases}$$

It can be shown that, unless $n \leq 2$, this condition cannot be satisfied if the distributions of $C_1, ..., C_n$ are independent.

Ted generates the outcomes of coins $C_1, ..., C_m$ inductively as follows (where the quantities $s_1, ..., s_m, t_1, ..., t_m \in [0,1]$ will be defined later). C_1 is a biased coin generated such that $\Pr[C_1 = S] = s_1$ and $\Pr[C_1 \neq S] = t_1$. Once $C_1, ..., C_i$ have been generated, C_{i+1} is generated according to the following distribution. For all $x_1...x_i \in \{0,1\}^i$,

$$\Pr[C_{i+1} = x_i | S = x_i \wedge C_1...C_i = x_1...x_i] = s_{i+1}$$

$$\Pr[C_{i+1} = x_i | S \neq x_i \wedge C_1...C_i = x_1...x_i] = t_{i+1}.$$

It can be verified that, if $C_1, ..., C_m$ are generated in this manner then, for all $x_1...x_{i+1} \in \{0,1\}^{i+1}$,

$$\Pr[S = x_1 | C_1 = x_1] = s_1 = 1 - t_1,$$

and, by applying Bayes' rule, for all $i \in \{1, ..., m-1\}$,

$$\Pr[S = x_{i+1} | C_1...C_{i+1} = x_1...x_{i+1}] = \begin{cases} \frac{p_i s_{i+1}}{p_i s_{i+1} + (1-p_i) t_{i+1}} & \text{if } x_{i+1} = x_i \\ \frac{(1-p_i)(1-t_{i+1})}{(1-p_i)(1-t_{i+1}) + p_i(1-s_{i+1})} & \text{if } x_{i+1} \neq x_i. \end{cases}$$

Therefore, in order to satisfy

$$\Pr[S = x_i | C_1 ... C_i = x_1 ... x_i] = p_i,$$

for all $i \in \{1, ..., m\}$ and for all $x_1 ... x_i \in \{0,1\}^i$, it is necessary and sufficient for $s_1, ..., s_m, t_1, ..., t_m$ to satisfy $s_1 = p_1$, $t_1 = 1 - p_1$, and, for all $i \in \{1, ..., m-1\}$,

$$\frac{p_i s_{i+1}}{p_i s_{i+1} + (1-p_i) t_{i+1}} = p_{i+1} = \frac{(1-p_i)(1-t_{i+1})}{(1-p_i)(1-t_{i+1}) + p_i(1-s_{i+1})}.$$

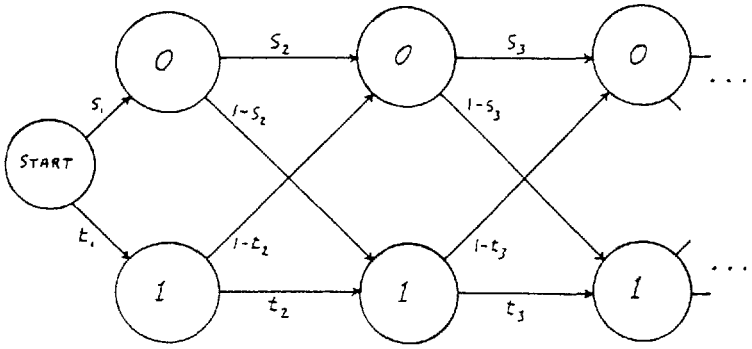
These equations yield a unique solution of $s_1 = p_1$, $t_1 = 1 - p_1$, and for all $i \in \{1, ..., m-1\}$,

$$s_{i+1} = \left(\frac{p_{i+1}}{p_i}\right)\left(\frac{p_i + p_{i+1} - 1}{2p_{i+1} - 1}\right)$$

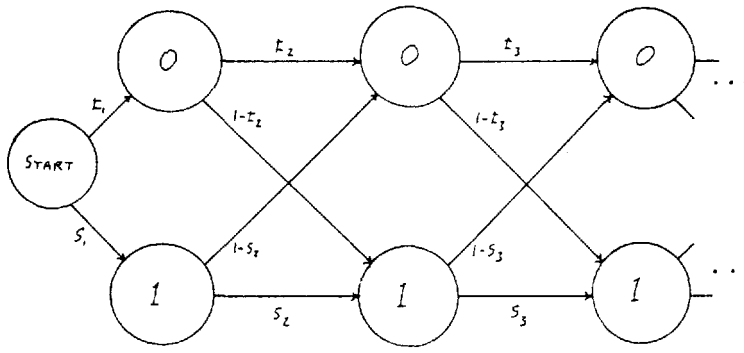
$$t_{i+1} = \left(\frac{1-p_{i+1}}{1-p_i}\right)\left(\frac{p_i + p_{i+1} - 1}{2p_{i+1} - 1}\right)$$

and, since $\frac{1}{2} < p_1 < \cdots < p_m = 1$, it can be verified that $s_1, ..., s_m, t_1, ..., t_m$ are all valid probabilities (i.e. they are all in the range $[0,1]$).

Intuitively, the values of $C_1, ..., C_m$ can be viewed as the states of the execution of a Markov chain. If Alice's secret is 0 then Ted selects an initial state to be 0 with probability s_1 and 1 with probability t_1, and shows Bob the states of an execution of the following Markov chain.



If, on the other hand, Alice's secret is 1 then Ted selects an initial state to be 1 with probability s_1 and 0 with probability t_1, and shows Bob the states of an execution of the following Markov chain.



Bob does not know which Markov chain is being executed but Bob does know that it is one of two (depending on what S is). Bob observes the sequence of states of the Markov chain that is executed and, from this and Bob's knowledge of probability theory, at stage i, infers that S is the current state of the Markov chain with probability p_i.

3.2 Second Implementation (without Trusted Third Party)

In this section, we give an overview of how Alice and Bob can cryptographically simulate the role that Ted plays in the protocol of Section 3.1. Essentially, what Alice and Bob are able to do is simulate the execution of the appropriate Markov chain for Alice's secret S with the following conditions holding: (1) (To protect the consistency of the secret) the Markov chain is certified to Bob to be the one that corresponds to S (explained in Section 3.1); (2) (To protect the security of the secret) Bob learns nothing about the Markov chain except what he can infer from the execution that he sees; (3) (To protect the obliviousness of the execution) Alice learns nothing about the particular execution of the Markov chain that Bob sees.

First, we design our protocol to satisfy conditions (2) and (3) under the assumption that Alice and Bob both correctly follow the protocol (but are then allowed to make inferences based on the messages that were exchanged during the protocol). (We later explain how to modify the protocol to satisfy (1), (2), and (3) when Alice and Bob are allowed to deviate from the protocol in an arbitrary manner.)

Our protocol relies on an implementation of a "one-out-of-k" oblivious transfer protocol (explained by Brassard, Crépeau and Robert [4]). This protocol enables Alice to set up k bits and Bob to read a bit of his choosing such that: Alice has no idea which bit Bob read; and Bob learns nothing about the other $k-1$ bits. Under the Quadratic Residuosity Conjecture, a one-out-of-k oblivious transfer protocol can be constructed as follows. Suppose the k bits are $b_1, ..., b_k$. Alice first generates two

distinct random n-bit primes, p and q, and then generates $d \in \mathbf{Z}'_{p \cdot q}$ such that d is a quadratic nonresidue. Then Alice independently generates random $x_1, ..., x_k \in \mathbf{Z}'_{p \cdot q}$ subject to x_i being a quadratic residue iff $b_i = 1$ ($i \in \{1, ..., k\}$) and Alice sends $p \cdot q, d, x_1, ..., x_k$ to Bob. (Note that, at this point, under the Quadratic Residuosity Conjecture, Bob cannot deduce anything about the value of any of the bits $b_1, ..., b_k$ from the information that he has.) Now, for Bob to learn bit b_{i_0}, he generates a random $y \in \mathbf{Z}'_{p \cdot q}$ and a random $t \in \{0, 1\}$ and sends $d^t \cdot y^2 \cdot x_{i_0}$ to Alice, who determines whether this quantity is a quadratic residue or not, and informs Bob of the result. From Alice's point of view, $d^t \cdot y^2 \cdot x_{i_0}$ is a random element of $\mathbf{Z}'_{p \cdot q}$, so she cannot deduce anything about the value of i_0 from this transaction. By the algebraic properties of $\mathbf{Z}'_{p \cdot q}$, Bob can infer whether x_{i_0} is a quadratic residue or not from the knowledge of whether $d^t \cdot y^2 \cdot x_{i_0}$ is a quadratic residue or not, and, thus, he can infer b_{i_0}. Also, it is straightforward to show that, under the Quadratic Residuosity Conjecture, Bob does not learn any information from this transaction about the value of the other b_is, ($i \neq i_0$).

Using a one-out-of-k oblivious transfer protocol, Alice can simulate a coin with a bias of her choosing and Bob can obtain an outcome of this coin so that: Alice does not see the outcome; and Bob learns nothing about the bias of the coin (except what he can infer from the outcome). To achieve this, Alice randomly chooses k bits subject to the condition that the proportion of 1s to 0s corresponds to the bias of the coin. To obtain an outcome, Bob randomly chooses $i \in \{1, ..., k\}$ and, using a one-out-of-k oblivious transfer protocol, determines the value of the i-th bit.

Using several of the above simulations of biased coins, Alice and Bob can appropriately simulate the execution of any Markov chain in which each state has at most two possible successors. To do this, Alice simulates a biased coin for each state in the Markov chain, where each of these simulations are in terms of elements of the same set $\mathbf{Z}'_{p \cdot q}$ (i.e. the p and q are the same for each coin). At each step, Bob obtains an outcome of the appropriate coin for the current state of the Markov chain to determine the next state (that is, he selects a random x_i from those that Alice sent him that correspond to the current state, and determines, through Alice, whether it is a quadratic residue or not). Since, at each stage, Alice only receives a random element of $\mathbf{Z}'_{p \cdot q}$ (and she determines its residue/nonresidue status and reveals this to Bob), Alice learns nothing about the states traversed in the specific execution of the Markov chain. Also, by the Quadratic Residuosity Conjecture, it can be shown that Bob learns nothing about the specific transition probabilities of the Markov chain (except what he can infer from the states traversed in its execution). Thus, under the assumption that Alice and Bob both correctly follow the protocol, conditions (2) and (3) are satisfied.

We now consider the general case where Alice and Bob may deviate from the protocol. For condition (1), Alice can initially send $p \cdot q$, d and several elements of $Z'_{p \cdot q}$ that represent the Markov chain to Bob and certify that they are valid for some S' by a zero-knowledge proof (explained by Goldreich, Micali, and Wigderson [6]). Also, during the further execution of the protocol, both Alice and Bob can certify that they are faithfully executing the protocol by zero-knowledge proofs. Since both Alice and Bob are constrained to probabilistic polynomial-time computations, all these zero-knowledge proofs can be implemented in a constant number of rounds.

4 Two-Party Secret Bit Exchanging Protocols

Suppose that parties Alice and Bob possess secret bits S_a and S_b (respectively) and that they would like to know each other's secrets and are willing to make an exchange of one secret for another. Suppose further that Alice and Bob do not trust each other in the following two senses. First, neither party is willing to believe that the other party is telling the truth about its secret—unless the other party *proves* the validity of the information that it sends. Also, each party is reluctant to reveal its secret first, for fear that the other party will not reveal its secret in return. This problem was investigated by Yao [13,14,15], Halpern and Rabin [9], Luby, Micali and Rackoff [10], and Tedrick [11]. Similar secret exchanging problems (except that they involve secrets that are multi-bit "keys") were investigated by Blum [2], Brickell, Chaum, Damgård, and van de Graaf [5], and Tedrick [11].

In this section, we give an informal sketch of how a controlled gradual disclosure scheme can be used to strengthen the security of the secret bit exchanging protocols proposed by Luby, Micali and Rackoff [10], and Yao in [15]. In all these protocols (as well as ours) it is assumed that, initially, each party presents a correct commitment of its secret to the other party.

4.1 Overview of Previous Results

In this section, we briefly review previous work on protocols for fair secret bit exchanges.

Halpern and Rabin [9] overcome the main difficulties of the problem by making the assumption in their model that, as soon as one party, say Alice, learns Bob's secret, Alice performs some action that Bob can observe. From this and what Bob has learned during the execution of the protocol, Bob is then able to infer what Alice's secret is.

In [14], Yao claims that in some contexts, a "fair" secret bit exchanging protocol does not exist. In [15], Yao claims that there exists a protocol for a generalized form

of secret exchanging in which the protocol of Luby, Micali and Rackoff [10], explained below, arises as a special case.

The protocol of Luby, Micali and Rackoff [10] is a secret exchanging scheme that satisfies the following properties. If both parties follow the protocol faithfully then, with very high probability, they successfully exchange secrets. If one party cheats then (under the quadratic residuosity conjecture) the amount of information that it obtains about the other party's secret is "fairly close" to the amount of information that the other party obtains about its secret. More precisely, if the total number of rounds in the protocol is $\omega(m^2 \log^2 m)$ then, after the execution of the protocol, the following holds. If both parties are honest then they can both guess each other's secrets with probability $1 - (\frac{1}{m})^{\omega(1)}$. If one party, say Alice, is honest and she can guess Bob's secret with probability p then Bob (even if he has cheated) can guess Alice's secret with probability bounded by $p + \frac{1}{m}$. (Note that, in this protocol, the discrepancy of $\frac{1}{m}$ between Alice's and Bob's knowledge cannot be made super-polynomially small (i.e. $(\frac{1}{n})^{\omega(1)}$) unless it runs for a super-polynomial number of rounds (i.e. $n^{\omega(1)}$). The lower bound that we show in Section 4.3 implies that no other protocol can achieve a super-polynomial discrepancy in a polynomial number of rounds.)

The protocol of Luby, Micali and Rackoff simulates a "symmetrically biased coin" that enables Alice and Bob to both gain a little knowledge (in a probabilistic sense) about each others secrets at each stage. Informally, if the secrets of Alice and Bob are S_a and S_b (respectively) then the symmetrically biased coin, C, is biased towards $S_a \oplus S_b$ by $\frac{1}{m}$. The protocol repeatedly "flips" C, each time enabling Alice and Bob, in turn, to see the outcome. Neither Alice nor Bob knows what the exact bias of C is (if one of them did, it would know $S_a \oplus S_b$ and could deduce the other party's secret). But seeing the outcome of a flip of C gives each party a little probabilistic information about the bias of C and, thus, the value of $S_a \oplus S_b$. Since each party also knows its own secret, each party as a result learns a little probabilistic information about the secret of the other party. As the coin C is flipped repeatedly, each party's expected knowledge of the other party's secret increases. After $\omega(m^2 \log^2 m)$ rounds, the expected confidences of the two parties about each others secrets are both $1 - (\frac{1}{m})^{\omega(1)}$. If one party cheats by quitting the protocol early, it only learns the outcome of at most one more coin flip of C than the other party learns. Thus, since C is only biased towards $S_a \oplus S_b$ by $\frac{1}{m}$, the discrepancies between the two parties confidence of their knowledge of each other's secrets is bounded by $\frac{1}{m}$.

During the execution of the above protocol, the pieces of knowledge that the two parties have about each other's secrets are highly correlated with each other. In fact, Alice and Bob's best guesses of each other's secrets after each stage are either both right or both wrong. Suppose that at some time before the execution of the protocol terminates, one party, say Alice, is confident about her knowledge of Bob's secret and

performs some "action in the outside world" based on what she thinks Bob's secret is, and Bob observes this action. Then Bob can infer Alice's secret in the following way. Bob knows his own secret and what Alice thinks Bob's secret is. If Alice is right then Bob knows that his current estimate of Alice's secret is right, otherwise Bob knows that his current estimate of Alice's secret is wrong. It is desirable for secret exchanging protocols to be secure against any such inferences. If Alice and Bob's knowledge about each other's secrets are independent of each other then no such inferences can be made. In the next section, we present a secret bit exchanging protocol, based on our controlled gradual disclosure scheme, that has this property and is thus secure against the aforementioned kinds of inferences.

It is important to note that the independent interleaving of two biased coin schemes (as described in Section 1) will yield a very insecure secret bit exchanging scheme. This observation (due to Luby, Micali and Rackoff [10]) is based on the fact that two sequences of probabilities that follow independent random walks will be very likely to, at some point, drift apart significantly. Thus, during the execution of such a protocol, there will likely be some point where there is a significant gap between Alice and Bob's respective knowledge about each other's secrets.

4.2 A Secret Bit Exchanging Protocol Based on a Controlled Gradual Disclosure Scheme

An m-round secret exchanging protocol can by constructed by interleaving two controlled gradual disclosure schemes, one in which Alice discloses her secret to Bob, and one in which Bob discloses his secret to Alice. The two executions of the controlled gradual disclosure scheme can be run independently in the sense that the honest parties use different random bits for the two protocols. If one party quits a protocol or does not send a valid message at some stage then the other party immediately quits both protocols and retains its current best estimate of the other party's secret.

This protocol has the property that if one of the players is honest then, during the execution of the protocol, the information that the parties have about each other's secrets are independent and, therefore the protocol is secure against the kinds of attacks discussed in Section 4.1.

Also, by setting $p_i = \frac{1}{2} + \frac{i}{2m}$ ($i \in \{1, ..., m\}$) in the controlled gradual disclosure schemes, we obtain the following property if one party cheats. After the execution of the protocol, if the honest party can guess the secret of the cheating party with probability p then the cheating party can guess the secret of the honest party with probability bounded by $p + \frac{1}{m}$ (thus, the discrepancy is $\frac{1}{m}$). The total number of rounds in the protocol is $O(m)$ (whereas the protocol of Luby, Micali and Rackoff [10] requires $\omega(m^2 \log^2 m)$ rounds to obtain the same discrepancy). In Section 4.3,

we prove that this relationship between discrepancy of information and number of rounds is asymptotically the best possible.

4.3 A Lower Bound on the Number of Rounds Required for a Secret Exchanging Protocol With a Given Level of Security

Theorem: *For any m-round secret exchanging scheme, one party can learn* $\Omega(\frac{1}{m})$ *more (in terms of its confidence level) about the other party's secret than it reveals about its own secret by quitting at an opportune time.*

Sketch of Proof: To understand the intuition behind this proof, consider the confidence of each party at intermediate stages during the execution of the protocol. Initially, these quantities are both close to $\frac{1}{2}$ and eventually they must approach 1 so the average "gap" between these confidence levels during the execution of the protocol is $\Omega(\frac{1}{m})$. A party that cheats by quitting where one of these gaps arises will have an advantage of $\Omega(\frac{1}{m})$. □

5 Multi-Party Function Evaluation Protocols

Suppose that Alice possesses bits $x_1, ..., x_k$ and Bob possesses bits $y_1, ..., y_k$ and that they are interested in learning the value of some function $f(x_1, ..., x_k, y_1, ..., y_k)$ through the execution of some protocol. Supposing that Alice and Bob do not trust each other, there are several notions of security that are of interest. One of these notions, called "fairness", is the property that one party, even by cheating during the execution of the protocol, cannot learn more about the value of $f(x_1, ..., x_k, y_1, ..., y_k)$ than the other party learns. A two-party version of this problem was considered by Yao [15], and a multi-party version is considered by Beaver and Goldwasser [1].

In this section, we very briefly review how the previous protocols handle the issue of fairness, and how a controlled gradual disclosure scheme can be used to make the protocols more efficient in the number of rounds that they require (and asymptotically optimal in this respect).

For simplicity, let us only consider the case where $f(x, y) = x \oplus y$ (so $k = 1$), and the prior distributions of x and y are independent and unformly random (the results that we discuss here extend to more general functions, and more general prior distributions of x and y).

Along the lines of the theorem in Section 4.3, it can be shown that any m-round protocol for this problem has the property that one party, by quitting an an opportune time, can obtain an "advantage" of $\Omega(\frac{1}{m})$ in the following sense. That party's

confidence level about the value of $f(x,y)$ exceeds the confidence level of the other party by $\Omega(\frac{1}{m})$.

The protocols proposed by Yao [15], and Beaver and Goldwasser [1] (in the two-party case), operate as follows. Alice and Bob jointly construct a coin C that is biased towards $f(x,y)$ by $\frac{1}{m}$, and, at successive stages, learn the outcome of independent flips of C. After $\omega(m^2 \log^2 m)$ stages, the expected value of Alice and Bob's confidence levels (about the value of $f(x,y)$) are $1-(\frac{1}{m})^{\omega(1)}$. If one party cheats by quitting the protocol early then the advantage of that party is at most the result of one more coin flip of C and is thus bounded by $\frac{1}{m}$. Note the gap between this result and the lower bound above: to bound the possible advantage of one party by $\frac{1}{m}$, $\Omega(m)$ rounds are necessary, while these protocols show that $\omega(m^2 \log^2 m)$ rounds are sufficient.

Using a controlled gradual scheme, an (asymptotically optimal) $O(m)$ round version of the above protocols is possible. This is achieved by (independently) interleaving the execution two controlled gradual schemes, one disclosing $f(x,y)$ to Alice, the other disclosing $f(x,y)$ to Bob.

6 Acknowledgments

Michael Luby and Charles Rackoff made many helpful remarks concerning this research.

References

[1] D. Beaver, and S. Goldwasser, "Multiparty Computation with Faulty Majority", *Advances in Cryptology — CRYPTO '89 Proc.*, these proceedings.

[2] M. Blum., "How to Exchange (Secret) Keys", *Proc. 15th Ann. ACM Symp. on Theory of Computing*, 1983, pp. 440–447.

[3] M. Blum, and S. Micali, "How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits", *Proc. 23th Ann. IEEE Symp. on Foundations of Computer Science*, 1982, pp. 112–117.

[4] G. Brassard, C. Crépeau, and J.-M. Robert, "Information Theoretic Reductions Among Disclosure Problems", *Proc. 27th Ann. IEEE Symp. on Foundations of Computer Science*, 1986, pp. 168–173.

[5] E. F. Brickell, D. Chaum, I. B. Damgård, and J. van de Graaf, "Gradual and Verifiable Release of a Secret", *Advances in Cryptology — CRYPTO '87 Proc.*, C. Pomerance (ed.), Lecture Notes in Computer Science **293**, Springer, 1988, pp. 156–166.

[6] O. Goldreich, S. Micali, and A. Wigderson, "Proofs That Yield Nothing But Their Validity and a Methodology for Cryptographic Protocol Design", *Proc. 27th Ann. IEEE Symp. on Foundations of Computer Science*, 1986, pp. 174–187.

[7] O. Goldreich, S. Micali, and A. Wigderson, "How to Play Any Mental Game", *Proc. 19th Ann. ACM Symp. on Theory of Computing*, 1987, pp. 218–229.

[8] S. Goldwasser, S. Micali, "Probabilistic Encryption & How to Play Mental Poker, Keeping Secret All Partial Information", *Proc. 14th Ann. ACM Symp. on Theory of Computing*, 1982, pp. 365–377.

[9] J. Y. Halpern, and M. O. Rabin, "A Logic to Reason About Likelihood", *Proc. 15th Ann. ACM Symp. on Theory of Computing*, 1983, pp. 310–319.

[10] M. Luby, S. Micali, and C. Rackoff, C., "How to Simultaneously Exchange a Secret Bit by Flipping a Symmetrically-Biased Coin", *Proc. 22nd Ann. IEEE Symp. on Foundations of Computer Science*, 1983, pp. 11–21.

[11] T. Tedrick, "How to Exchange Half a Bit", *Advances in Cryptology: Proc. of CRYPTO '83*, D. Chaum (ed.), Plenum, 1984, pp. 147–151.

[12] U. Vazirani, and V. Vazirani, V., "Trapdoor Pseudo-Random Number Generators, With Applications to Protocol Design", *Proc. 22nd Ann. IEEE Symp. on Foundations of Computer Science*, 1983, pp. 23–30.

[13] A. Yao, "Theory and Applications of Trapdoor Functions", *Proc. 21st Ann. IEEE Symp. on Foundations of Computer Science*, 1982, pp. 80–91.

[14] A. Yao, "Protocols for Secure Computations", *Proc. 21st Ann. IEEE Symp. on Foundations of Computer Science*, 1982, pp.160–169.

[15] A. Yao, "How to Generate and Exchange Secrets", *Proc. 25th Ann. IEEE Symp. on Foundations of Computer Science*, 1986, pp. 162–167.