Information Theoretic Reductions among Disclosure Problems

Gilles BRASSARD† Claude CREPEAU ‡Jean-Marc ROBERT
Département d‘I.R.O.
Université de Montréal

Department of Computer Science
MIT

Département de Génie Électrique
Ecole Polytechnique de Montréal

Abstract

Alice disposes of some number of secrets. She is willing to disclose one of them to Bob. Although she agrees to let him choose which secret he wants, she is not willing to allow him to gain any information on more than one secret. On the other hand, Bob does not want Alice to know which secret he wishes. An all-or-nothing disclosure is one by which, as soon as Bob has gained any information whatsoever on one of Alice’s secrets, he has wasted his chances to learn anything about the other secrets. We assume that Alice is honest when she claims to be willing to disclose one secret to Bob (i.e. she is not about to send junk). The only cheating Alice is susceptible of trying is to figure out which secret is of interest to Bob. We address the following question from an information theoretic point of view: what is the most elementary disclosure problem? The main result is that the general all-or-nothing disclosure of secrets is equivalent to a much simpler problem, which we call the two-bit problem.

1. INTRODUCTION

Alice disposes of some number of secrets. She is willing to disclose one of them to Bob. Although she agrees to let him choose which secret he wants, she is not willing to allow him to gain any information on more than one secret. On the other hand, Bob does not want Alice to know which secret he wishes. This is a useful building block in crypto-protocols. For instance, it can be used to easily implement a multi-party mental Poker protocol similar to that of [C1], i.e.: safe against player coalitions. An all-or-nothing disclosure is one by which, as soon as Bob has gained any information whatsoever on one of Alice’s secrets, he has wasted his chances to learn anything about the other secrets. In particular, it must be impossible for Bob to gain joint information on several secrets, such as their exclusive-or. Notice that this is crucial, because it is well-known in classical cryptography that the exclusive-or of two plaintext English messages allows easy recovery of them both, just as a running stream Vigenère would [D].

We assume that Alice is honest when she claims to be willing to disclose one secret to Bob (i.e. she is not about to send junk). The only cheating Alice is susceptible of trying is to figure out which secret is of interest to Bob. Although equally worthwhile, we do not address here the problem of verifiable secrets, because it is too much application dependent and because it does not make sense in our information theoretic setting in which both parties could have unlimited computing power.

Let us stress that the major novelty consists in Bob’s choosing which secret he obtains. This is interesting whenever the secrets are not anonymous: although Bob does not know their contents, he knows their individual purpose. Consider for instance the following situation: an international spy disposes of a large corpus of various state secrets. He sells them by the piece to whoever is willing to pay the price. In his catalogue, each secret is advertised with a tantalizing title, such as “where is Abu Nidal”. He would not accept to give away two secrets for the price of one, or even partial information on more than one secret. On the other hand, you (the potential buyer) would not pay for a randomly chosen secret, but are reluctant to let him know which secret you wish to acquire, because his knowledge of your specific interests could be a valuable secret for him to sell to someone else (under the title: “who is looking for terrorists”, for instance). Let us point out that this problem was addressed and solved more than 15 years ago by quantum physical means, when the number of secrets is at most three, in Wiesner’s original Quantum Cryptography paper [W].

Under cryptographic assumptions, a practical computationally secure protocol for this problem has been proposed in [BCR]. It is reviewed at the end of this paper. Notice that, for its particular application in [C2] (mental poker) this protocol forces Alice to cooperate (i.e. the secrets are verifiable). Here, we address the following question from an information theoretic point of view: what is the most elementary disclosure problem? It turns out that the general all-

† Supported in part by NSERC grant A107.
‡ Supported in part by an NSERC postgraduate scholarship.
* Research conducted at the Université de Montréal

0272-5428/86/0000/0168$01.00 © 1986 IEEE
or-nothing disclosure of secrets is equivalent to the two-bit problem (described below). This result does not depend on computational complexity cryptographic assumptions.

2. THE MOST ELEMENTARY DISCLOSURE PROBLEM

It is exactly as hard to all-or-nothing disclose one \( t \)-bit secret among \( n \) than it is to disclose one bit among two. This result is obtained by a chain of reductions that allows the collapse of an apparent hierarchy of disclosure problems. Here is a list of problems that turn out to be information-theoretically equivalent, that is even if either or both party(ies) had unlimited computing power.

The two-bit problem (2BP): Alice disposes of two secret bits and she is willing to disclose one of them to Bob, at his choosing. Bob must not be allowed to learn more than one bit of information on Alice’s bits, but Alice will not be upset if Bob succeeds in gaining any (deterministic) one-bit function of these two bits, such as their exclusive-or. If Bob plays fair and obtains the physical bit of his choice, Alice does not know which of her two bits she disclosed.

The all-or-nothing two-bit problem (AN2BP): Alice disposes of two secret bits and she is willing to disclose one of them to Bob, at his choosing. Nothing Bob can do will give him more than one of these physical bits: as soon as he obtains any information on one of them, he loses all hopes to gain any information on the other. Alice does not know which of her two bits she disclosed.

The all-or-nothing n-bit problem (ANNBP): This is identical to the previous problem, except that Alice owns \( n \) secret bits rather than 2. She wishes to all-or-nothing disclose one of them to Bob, at Bob’s choosing.

The all-or-nothing disclosure of secrets (ANDOS): Described previously.

We shall now sketch how to efficiently transform any protocol for 2BP into one for AN2BP, any protocol for AN2BP into one for ANNBP, and any protocol for ANNBP into one for ANDOS. The most interesting reduction is the last one, so that we will be rather brief about the first two. More details will be provided in the final paper. The first reduction (2BP \( \Rightarrow \) AN2BP) allows an exponentially small probability of undetected cheating, but no amounts of computing power could increase this probability. The other two reductions, however, are information-theoretically perfect in the sense that any fool-proof solution to AN2BP would yield a fool-proof solution to the general ANDOS problem. An information-theoretic secure solution to any of these problems, including the elementary 2BP, would be of considerable interest. Under cryptographic assumptions, we review the computationally secure solution of [BCR] in section 4.

2.1. 2BP \( \Rightarrow \) AN2BP

Assume the availability of a protocol for 2BP. Let Alice dispose of two secret bits \( a \) and \( b \), of which she is willing to all-or-nothing disclose one to Bob.

protocol 1

Let \( m \) be an even integer, used as safety parameter. Alice randomly chooses a subset \( X \subseteq \{1, 2, \ldots, m\} \) of size \( m/2 \). Let \( u \) and \( v \) be the smallest positive integers within \( X \) and outside \( X \), respectively. Alice randomly chooses a total of \( 2m-2 \) bits \( \{r_i, s_i\} \), \( 1 \leq i \leq m \), \( i \neq u \) and \( 1 \leq j \leq m \), \( j \neq v \). Let \( m \) be the smallest positive integers within \( X \) and outside \( X \), respectively. Alice randomly chooses a total of \( 2m-2 \) bits \( \{r_i, s_i\} \), \( 1 \leq i \leq m \), \( i \neq u \) and \( 1 \leq j \leq m \), \( j \neq v \). She sets the bits \( r_u \) and \( s_v \) such that \( a = \{r_i \mid i \in X\} \) and \( b = \{s_i \mid i \notin X\} \), where ‘\( @ \)’ denotes the exclusive-or. She uses the 2BP protocol to disclose Bob one of \( r_1 \) or \( s_1 \), one of \( r_2 \) or \( s_2 \), and of \( r_m \) or \( s_m \). Only then does she give \( X \) to Bob. If he systematically asked for all \( r_i \)'s (resp. all \( s_i \)'s), he can easily reconstruct \( a \) (resp. \( b \)).

This protocol allows Bob to attempt cheating with a non-zero probability of success: if he chooses randomly to read \( m/2 \) of the \( r_i \)'s and \( m/2 \) of the \( s_i \)'s, he obtains both \( a \) and \( b \) only if he guessed \( X \) correctly, which happens with probability \( 1/\binom{m}{m/2} \approx 2^{-m^{1/2}} \). With such a strategy, however, he has an overwhelming probability to get absolutely no information on neither \( a \) nor \( b \). Another strategy would be to ask for \( r_i \oplus s_i \) for some \( i \); but this is dumb because it irrevocably wastes his chances to learn one of \( a \) or \( b \), depending of whether \( i \in X \). The analysis of this protocol becomes significantly harder if Bob attempts biased questions on \( r_1 \) and \( s_1 \), such as their conjunction, for some values of \( i \). In this case, use of Bernstein’s Law of Large Numbers [K] allows us to prove that, no matter which \( m \) bits he requests from the 2BP protocol, Bob only has an exponentially small chance of getting more than an exponentially small advantage on both \( a \) and \( b \), simultaneously. The details are quite messy; they can be found in the final version of this paper.

2.2. AN2BP \( \Rightarrow \) ANNBP

Assume the availability of a protocol for AN2BP. Let Alice dispose of \( n \) secret bits \( b_1, b_2, \ldots, b_n \), of which she is willing to all-or-nothing disclose one to Bob.

protocol 2

Alice randomly chooses \( n-2 \) bits \( r_1, r_2, \ldots, r_{n-2} \). She then uses the AN2BP protocol to all-or-nothing disclose Bob one of \( b_1 \) or \( r_1 \), one of \( b_2 \oplus r_1 \) or \( r_2 \oplus r_1 \), \ldots, one of \( b_{n-2} \oplus r_{n-3} \) or \( r_{n-2} \oplus r_{n-3} \), and one of \( b_{n-1} \oplus r_{n-2} \) or \( b_n \oplus r_{n-2} \). If Bob wishes secret \( b_1 \), he simply asks for it in the first instance of the AN2BP protocol. If Bob wishes secret \( b_s \), for \( 2 \leq s \leq n-1 \), he asks
Therefore, in order to obtain Bob to.

Let Alice dispose of an integer $m$. This gives

"L: {0,1}m be the Boolean matrix associated

Theorem. Let $F$ be the Boolean matrix associated

Proof (sketch).

Necessary: if the condition fails with $X = Y$, the exclusive-or of the $X$-bits of $f(x)$ is always zero, hence the empty set is in the information support of $f$, so that $f$ cannot be a zigzag; if it fails with $X \neq Y$, the exclusive-or of the $X$-bits of $f(x)$ and the exclusive-or of the $Y$-bits of $f(y)$ can both be obtained by asking disjoint questions on bits of $x$ and $y$.

Sufficient: this is a consequence of the xor-lemma of [BBR], which says in essence that any partial information on $f(x)$ obtained by knowledge of specific bits of $x$ automatically gives complete knowledge on some exclusive-or of the bits of $f(x)$.

This characterisation allows to prove by induction the existence of a linear $(m,t)$-zigzag whenever $m = 3^{\log_2 t}$. For any integer $k \geq 0$, recursively define the $2^k \times 3^k$ matrix $F_k$ as $F_0 = [1]$, and

$$F_{k+1} = \begin{bmatrix} F_k & 0_k & F_k \\ 0_k & F_k & F_k \end{bmatrix}$$

where $0_k$ is the identically zero $2^k \times 3^k$ matrix. For any $t$, let $k = \lceil \log_2 t \rceil$. Any $t$ rows of $F_k$ defines a linear $(3^k,t)$-zigzag.

For instance,

$$F_2 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

defines a $(9,4)$-zigzag. It is clear that $m$ is polynomial in $t$ ($m = O(t^{1.59})$), this zigzag is easy to compute, and random inverses are easy to select. This completes the information-
theoretic reduction from any efficient solution for the two-bit problem to an efficient solution for the all-or-nothing disclosure of secrets problem.

There is a nice graphical representation of this construction that explains why we call such functions "zigzags". The figure in the Appendix shows a (27,8)-zigzag. The nodes on the left correspond to the 27 input bits, whereas the nodes on the right correspond to the 8 output bits. Each non-input node should be thought of as computing the exclusive-or of its two inputs. Given output x, this suggests a systematic way to randomly select an input y such that 

\[ x = f(y) \]  

peel off level by level, from right to left.

The optimality question for zigzags is still open: could there exist an \( (m,2^k) \)-zigzag for \( m < 3^k \)? One very crude lower bound on \( m \) for the existence of linear \( (m,t) \)-zigzags can be obtained by a reduction from the theory of binary linear error-correcting codes [MS]. Indeed, the matrix for any linear \( (m,t) \)-zigzag must be the generator matrix of an \([m,t]\) binary linear code such that the minimal distance between any two codewords is at least \( t \) (the converse fails in general). Therefore, from Griesmer's bound [G], we get 

\[ m \geq 3\tau - 2 - \beta(\tau - 1) \]  

where \( \beta(x) \) is the number of 1-bits in the binary representation of \( x \). There could nonetheless exist a more efficient non-linear zigzag.

After reading a first draft of this paper, Oded Goldreich and Silvio Micali discovered a different characterization for zigzags. Consider a \( t \times m \) matrix \( F \) and some set \( I \subseteq \{1,2, \ldots, m\} \). Let \( F_I \) denote the \( t \times \#I \) submatrix consisting of the \( I \)-columns of \( F \) and let \( F^I \) denote the \( t \times (m-\#I) \) submatrix consisting of \( F \) with its \( I \)-columns removed. The function defined by matrix \( F \) is an \( (m,t) \)-zigzag if and only if, for every \( I \subseteq \{1,2, \ldots, m\} \), at least one of \( F_I \) or \( F^I \) has full rank. More intuitively, 

\[ f : \{0,1\}^m \rightarrow \{0,1\}^t \]  

is an \( (m,t) \)-zigzag if and only if "for every set of indices \( I \subseteq \{1,2, \ldots, m\} \) either (for every \( b \)) \( x \)'s bits indexed by \( I \) are not determined given \( f(x) = b \) or this holds for \( x \)'s bits indexed by \( \{1,2, \ldots, m\} - I' \) [GrM].

Umesh Vazirani has suggested a different approach for the \( \text{ANNBP} \Rightarrow \text{ANDOS} \) reduction [V]. His insight was to use again the idea behind the \( 2\text{BP} \Rightarrow \text{AN2BP} \) reduction of Section 2.1. This leads to a more economical protocol for \( \text{ANDOS} \), at the cost of introducing an exponentially small probability of undetected cheating even if the \( \text{ANNBP} \) protocol had been perfect.

3. RELATED RESULTS (sketch)

There are two intermediate problems between \( \text{ANNBP} \) and \( \text{ANDOS} \) when Alice disposes of \( n \) \( t \)-bit secrets, one of which she is willing to disclose to Bob. In \( \text{ANDOS} \), she does not want Bob to get information on more than one secret. One natural intermediate problem is for her to tolerate this kind of cheating, as long as Bob does not get physical bits from more than one secret. The other intermediate problem is more restrictive: Alice wants to make sure that if Bob ever gets a physical bit from some secret, he cannot get information on any other secret (therefore, he could get information on more than one secret only if he were willing to give up knowledge of physical bits altogether).

We have simple solutions for both these problems; if we redefine the notion of zigzag accordingly, we have a \((t,r)-\text{neo1-zigzag} \) and a \((2t-1,r)-\text{neo2-zigzag} \) for the reduction of \( \text{ANNBP} \) to each of these two intermediate problems. Both these reductions are proven optimal among all linear schemes. As a consequence of the lower bound mentioned at the end of the previous section, the all-or-nothing disclosure of secrets problem is strictly harder than both intermediate problems, assuming we only consider linear reductions from the all-or-nothing \( n \)-bit problem.

4. OUTLINE OF THE SCHEME OF [BCR]

Because the proceedings of the CRYPTO conference may not be widely distributed this year (1986), let us describe the quadratic residuosity based computationally secure \( \text{ANDOS} \) protocol given in [BCR]. We assume here that the reader has some number theoretic background, being familiar with the notation \( \mathbb{Z}_m^* \), the notions of quadratic residues and Jacobi symbols, and the quadratic residuosity assumption (QRA) [GwM]. We also assume the reader is familiar with the principle of zero-knowledge interactive proofs [GMR, GHY, BC].

Let \( x_1, x_2, \ldots, x_n \) be Alice’s \( t \)-bit secrets, and let \( b_{ij} \) be \( x_j \)'s \( j^{th} \) bit for \( 1 \leq i \leq n \) and \( 1 \leq j \leq t \). Initially, Alice randomly selects two distinct large primes \( p \) and \( q \) together with a quadratic non-residue \( y \) modulo \( m = pq \) whose Jacobi symbol is \(+1\). For each secret bit \( b_{ij} \), she selects a random \( x_{ij} \in \mathbb{Z}_m^* \) and computes \( z_{ij} = x_{ij}^2 y^{b_{ij}} \mod m \). Notice that \( z_{ij} \) is a quadratic residue if and only if \( b_{ij} = 0 \). Finally, Alice gives Bob both \( m \) and \( y \), keeping \( p \) and \( q \) secret, together with all the \( z_{ij} \)'s. According to QRA, this does not enable Bob to obtain in polynomial time any information on Alice’s actual secrets.

If Bob wanted to know bit \( b_{ij} \) for one specific \( i \) and \( j \), and if Alice were willing to cooperate, the following protocol comes to mind: Bob chooses a random \( r \in \mathbb{Z}_m^* \) and a random bit \( a \), he computes the question \( q = z_{ij} r^2 y^a \mod m \) and he asks Alice for the quadratic residuosity of \( q \). Clearly, \( b_{ij} = a \) if and only if \( q \) is a quadratic residue. On the other hand, regardless of \( i \) and \( j \), \( q \) is a completely random element of \( \mathbb{Z}_m^* \) and thus Alice has no idea as to which of her secret bits she gave away. One might be tempted to "solve" \( \text{ANDOS} \) by allowing Bob to ask \( t \) such questions, one for each bit of the secret he wants. There are three severe flaws with this idea:

- Bob could ask for \( t \) bits taken from distinct secrets.
- Bob could obtain in one question the exclusive-or of several bits. For instance, he could ask the question
\( q = z_{ij}^2 + \gamma^q \mod m \) and therefore learn \( b_{ij} \oplus b_{kj} \).

As pointed out in the introduction, this would most probably enable him to obtain two complete secrets by asking for their exclusive-or, assuming the actual secrets are in plaintext.

- More subtly, despite the previous claim, this would open the door for Alice to cheat as well! Indeed, she could lie from the beginning and give Bob a quadratic residue for her \( y \). In this case, the questions asked by the unsuspecting Bob would keep the same quadratic character as the corresponding \( z \)'s, allowing Alice to figure out Bob's interests.

In order to solve these difficulties, it is imperative that both Alice and Bob convince the other of her/his good faith: Alice must show that the information she posted initially is genuine and Bob must convince Alice that his questions are honest. This is where zero-knowledge interactive protocols come into play. The third problem above is solved by Alice using zero-knowledge interactive protocols of [GHY] and [GMR] to convince Bob that \( m \) has only two prime factors and that \( y \) is a quadratic non-residue modulo \( m \), respectively.

In a context of verifiable secret, this is where Alice would also convince Bob that the secrets hidden by the \( z_{ij} \)'s respect whichever conditions beft the application (a specific example is given in [C2]).

The first two problems above are harder to control. Although we have found several solutions, we only sketch here our favourite. Let \( \sigma \) be a permutation of \{1, 2, \ldots, n\}.

A \( \sigma \)-packet \( P_\sigma \) consists of one question for each bit of each secret in the following way: \( P_\sigma = \{q_{ij} \mid 1 \leq k \leq n, 1 \leq j \leq t\} \) such that each \( q_{ij} = z_{ij} r_{ij}^2 \gamma^h \mod m \), where \( r_{ij} \) is a random element of \( \mathbb{Z}_m^* \), \( r_{ij}^2 \) is a random bit and \( h = \sigma^{-1}(k) \). Moreover, a \( \sigma \)-packet is valid if Bob knows the corresponding \( \sigma \), \( r_{ij}^2 \)'s and \( q_{ij} \)'s (notice that any collection of \( nt \) elements of \( \mathbb{Z}_m^* \) is a \( \sigma \)-packet for every permutation \( \sigma \), and any valid packet looks like any other collections of random elements of \( \mathbb{Z}_m^* \) to Alice).

After the initialisation described previously, the ANDOS protocol proceeds as follows if \( x_i \) is the secret of interest to Bob.

- Bob selects a random permutation \( \sigma \) and forms a valid \( \sigma \)-packet \( P_\sigma \).
- Bob gives \( P_\sigma \) to Alice, keeping \( \sigma \) secret, and convinces her that it is valid (see below).
- Bob sends \( k = \sigma(i) \) to Alice as his actual request.
- Alice gives Bob the quadratic character of each \( d_{ij} \) in Bob's packet \( P_\sigma \), for this specific \( k \) and each \( 1 \leq j \leq t \).
- Bob infers each of Alice's bits \( b_{ij} \) for \( 1 \leq j \leq t \), hence he obtains \( x_i \) as desired.
- If Bob wishes to obtain another secret and if Alice is willing to give (or sell) it to him, it suffices to repeat the previous 3 steps with the relevant new value for \( i \).

It is of course crucial that Alice be convinced that Bob's packet is valid, for he could otherwise stuff it with dishonest questions and we would be back to the beginning. This is achieved by an idea very similar to those leading to the perfect zero-knowledge interactive protocol of [BC]. Let \( s \) be a safety parameter agreed upon between Alice and Bob. After giving Alice his \( \sigma \)-packet \( P_\sigma \), Bob chooses \( s \) additional permutations \( \sigma_1, \sigma_2, \ldots, \sigma_s \) of \{1, 2, \ldots, n\} and he creates \( s \) additional \( \sigma \)-packets \( P_1, P_2, \ldots, P_s \). He sends all these packets together with the original \( P_\sigma \). At this point, Alice selects a random subset \( X \subseteq \{1, 2, \ldots, s\} \) and sends it to Bob as a challenge. In order to convince her of the validity of \( P_\sigma \), Bob must:

- for each \( i \in X \), prove the validity of \( P_i \) to Alice by disclosing \( \sigma_i \) and all the random elements of \( \mathbb{Z}_m^* \) and random bits used in the creation of \( P_i \);
- for each \( i \not\in X \), prove to Alice that \( P_i \) is valid if and only if \( P_i \) is valid by disclosing \( \sigma_i \sigma_i^{-1} \) and showing that he is capable of transforming the questions in \( P_i \) into the corresponding questions in \( P_i \) (we leave the details of this to the reader).

At the end of this subprotocol, Alice will be convinced that \( P_\sigma \) is valid, with a \( 2^{-s} \) probability of being fooled by Bob. Indeed, the only way he could convince her her of the validity of an invalid \( P_\sigma \) would be by producing valid packets for each \( i \in X \) and invalid packets for each \( i \not\in X \). Since he must do so before being told \( X \), the result follows from the fact that Alice has \( 2^s \) different choices for \( X \).

ACKNOWLEDGEMENTS

We wish to thank David Chaum, Oded Goldreich, Silvio Micali, René Peralta, Joel Seiferas and Umesh Vazirani for their valuable comments on our protocols and for suggesting improvements. Isabelle Duchesnay has suggested the international spy application. Gilles Brassard also wishes to thank Manuel Blum for his inspiring talk at the MIT Endicott House conference in June 1985.

REFERENCES


