Blind Sales in Electronic Commerce

E. Aïmeur, G. Brassard and F. S. Mani Onana
Université de Montréal
Département d'informatique et de recherche opérationnelle
CP 6128, Succ. Centre-Ville
Montréal (Québec), H3C 3J7 Canada
{aimeur,brassard,manionaf}@iro.umontreal.ca

ABSTRACT

We start with the usual paradigm in electronic commerce: a consumer who wants to buy from a merchant. However, both parties wish to enjoy maximal privacy. In addition to remaining anonymous, the consumer wants to hide her browsing pattern and even the identification of the product she may decide to buy. Nevertheless, she wants to be able to negotiate the price, pay, receive the product and even enjoy maintenance on it. On the other hand, the merchant wants to leak as little information as possible on his catalogue for fear that he might in fact be dealing with a hostile competitor. For this purpose, we introduce the *Blind Customer Buying Behaviour* model, which adds confidentiality to the standard Customer Buying Behaviour model. In this paper, we concentrate on blind catalogue browsing.

Keywords

Electronic Commerce, Customer Buying Behaviour, Cryptography, Oblivious Transfer, Private Information Retrieval, CAPTCHA, Anonymous Surfing.

1. INTRODUCTION

Electronic commerce (EC or e-commerce) is concerned with the buying, selling and exchanging of products (goods and services) and the handling of transactions over telecommunication networks, such as the Internet [39]. The buying and selling bind two entities, the consumer and the merchant, herein referred to as Alice and Bob, respectively. Alice can be any individual or group of people who wish to buy products sold by Bob. She operates from anywhere with a computer or by any means (the Internet, phone, mail, etc.) that allows her to communicate with Bob. As for Bob, he is any individual or group of people who have products to sell through a telecommunication network.

The e-commerce process is better defined through six stages, known as the *Consumer Buying Behaviour* (CBB) model of Guttman, Moukas and Maes [20].

ICEC'04, Sixth International Conference on Electronic Commerce
Edited by: Marijn Janssen, Henk G. Sol, and René W. Wagenaar

Need identification: It is either a need or a kind of stimulation for the consumer, Alice, who gets general or personalized information from web advertisement (including, unfortunately, spams). If she is registered in an advertisement or a mailing list, at `amazon.com` for example, she may receive requested email notifications. Within this stage, it is possible to generate interest profiles through online subscriptions.

Product brokering: Here, there is a matching of Alice's need with offered products. In other words, this stage is concerned by what Alice wants to buy.

Merchant brokering: At this stage, Alice is looking for a certain merchant, say Bob, from whom to buy. She is assumed to have access to a directory of markets, merchants and products.

Negotiation: Within this stage, Alice and Bob use strategies and utility functions in order to reach an agreement. This stage can be highly interactive.

Payment and delivery: This stage is about the logging of transactions, the interfaces to online payment systems and the online delivery.

Service and evaluation: It is the feedback functionality that helps evaluate the quality of the *bond* between the consumer and the merchant. This stage is also about after-sale service and maintenance.

In the CBB model, Alice allows Bob to collect information about her, whether or not she so desires, at each of the six stages. In particular, Bob has mechanisms [39] to get Alice's *profile*, which is a portrait of who she is (age, gender, nationality, ethnic group, marital status, number of children, residence, income, education, buying behaviour, values, personality, tastes, interests, hobbies, etc.), as well as her Web browsing behaviour. The primary purpose of this paper is to offer cryptographic tools to fight in behalf of Alice against profile creation or dossier constitution [10], but also to protect confidential information in Bob's catalogue, while keeping the spirit of all six stages of the CBB model.

PROBLEM STATEMENT: In the context of e-commerce, the merchant, Bob, often has a lot of information about the consumer, Alice, so that he can use it for himself or share it with other merchants and organizations. Our objective is to

help Alice buy products without revealing her profile. More precisely, Bob's catalogue is a database $Y = [Y_1, Y_2, \ldots, Y_t]$ of the products he is selling. Alice wants to find and eventually buy a product Y_i, for some $1 \leq i \leq t$, that satisfies her requirements, which are described by the characteristics of the product she wants, such as: name, colour, width, length, height, etc.

We do not want Alice to send her complete request to Bob straight away, but rather to send many subqueries and to adapt these subqueries as she learns about the existence and availability of the product she seeks. For example, if Alice wants to buy a *red shirt of size 42*, she can proceed as follows. At first, she *blinds* the query "Do you have a shirt?" for Bob. By "blinding", we mean that Bob will not be able to know what the actual query is (see Section 4.2). Nevertheless, he will be able to send the response "I have it" (should he indeed have shirts to sell). Then, Alice continues with blind queries "Do you have a *red* shirt?" and "Do you have a red shirt *of size 42*?", which Bob may also answer by the affirmative. Now, Alice should be satisfied with the characteristics she is looking for. Only at this point will she be allowed to query the selling price of that "red shirt of size 42", in which case it will become impossible for her to change the characteristics of the desired product. This ensures that Alice will learn the price of only one product from Bob's catalogue. But as long as the selling price is not queried by Alice, Bob allows her to readjust the characteristics of the product she wants to buy. For example, if he does not have a *red* shirt, Alice may choose another colour. However, Bob will force Alice to prove that she is a human, rather than a robot, by submitting her at regular interval to a CAPTCHA [40]—see Section 3.4. This serves to prevent a hostile competitor from plundering Bob's catalogue by rapid-fire requests.

When the selling price is queried by Alice, Bob responds in a way that allows her only—but not him!—to learn the price (see Section 4.2 again). The fact that Bob remains ignorant of the price makes it impossible for him to guess what product Alice was seeking. At this point a *negotiation phase* could take place, by which Alice would negotiate the price so that Bob has to reduce it or lose the sale, but this must still be done according to the blind paradigm. At the end of the negotiation, Alice can decide to buy the product or to cancel her order. If Alice decides to buy the product, she pays by making use of an untraceable payment system [10]. In case of a *digital* product (such as a song or a piece of software), she receives it electronically through an untraceable communication link [9]. Moreover, she should be able in the future to obtain updates and corrective measures, if appropriate, still without ever revealing to Bob who she is or what she has bought.

We note that Alice's privacy is emphasized more than Bob's. Indeed, Alice's identity remains anonymous throughout the process and Bob cannot learn anything about what she browsed in his catalogue and the final product she may decide to buy. On the other hand, partial information on Bob's catalogue leaks to Alice since she can learn the availability (but not the price) of several items. By repeat visits, a determined Alice can in fact access the entire contents of Bob's catalogue. The fact that this cannot be prevented is the price to pay for Alice's anonymity, but it would be very laborious since the use of CAPTCHAs makes it impossible for Alice to program a robot to do the work for her. These issues are addressed in Section 4.1.

CONTRIBUTIONS: In this paper, we propose a *Blind Customer Buying Behaviour* (BCBB) model. Using the BCBB model, we offer a buying/selling protocol in which Alice's demands and Bob's offers are blinded in such a way that Bob learns no information about Alice's identity or her need, and Alice learns the selling price of one item only in Bob's catalogue. At the end of the protocol, Alice knows if Bob has the product she was looking for (or a replacement product), and she obtains complete information about that product, including its price. Alternatively, she could learn that the product she was requesting does not exist in Bob's catalogue, or that it is in backorder. Moreover, if Alice buys a *digital* product, Bob will not know what he sold (provided he has more than one digital product to sell, of course). Later, Alice can download updates or corrective measures as they become available. To define our BCBB model, we follow the standard CBB model, but we regroup its first three stages into one:

1. **Blind Search (BliS)**: stages 1 to 3 in the CBB model [20];
2. **Blind Negotiation (BliN)**: stage 4 in the CBB model;
3. **Blind Payment (BliP)**: stage 5 in the CBB model;
4. **Blind Maintenance (BliM)**: stage 6 in the CBB model.

Even though we define four stages in our BCBB model, we concentrate on the first stage in this paper. We briefly sketch solutions for the other three stages, but we defer their thorough treatment to a follow-up paper.

OUTLINE: This paper is organized as follows. In Section 2, we present the motivation for this work. In Section 3, we review some known results (related work) as well as various cryptographic preliminaries that we shall need to implement all four stages of the BCBB model. In Section 4, we present a detailed solution for the Blind Search (BliS) problem, which uses ElGamal's cryptosystem [16], and we sketch techniques to manage the other three stages of the BCBB model: Yao's *Millionaire's Problem* [42] is used for the Blind Negotiation (BliN) protocol, Chaum's untraceable payment system [10] combined with the "Priced Oblivious Transfer" of Aiello, Ishai and Reingold [1] are used for the Blind Payment (BliP) protocol, and ElGamal's cryptosystem is used again for the Blind Maintenance (BliM) protocol. We conclude and present perspectives for future work in Section 5.

2. MOTIVATION

The objective of this work is twofold. First, we eliminate the possibility for Bob to compile Alice's profile and second, we restrict Alice to learning only one entry from Bob's catalogue. Our first objective is greatly inspired from Chaum's seminal fight against Big Brother [10].

Chaum is the precursor of the notion of transactions without identification. More precisely, he proposed an approach to protect information about individuals that private and public organizations usually exchange for many purposes such as statistics, data mining, creating individual's profile, etc. His approach also protects organizations against possible abuses from individuals. Chaum's approach is applied in three kinds of consumer transactions: *communication*, *payments* and *credentials*. For the communication transactions, Chaum proposed a system in which an individual gives different pseudonyms to each organization with which he does business, so that it is impossible for organizations to link records from different sources and then generate dossiers on him. The payment transactions allow an individual to pay for products with untraceable electronic cash. As for the credential mechanisms, they allow an individual to prove to an organization that he has the required credentials without disclosing additional information on his set of credentials.

In Section 1, we reviewed the classic CBB model. Our work aims at eliminating any information that Bob could gain from Alice before she decides to buy (stages 1 to 3), while she is buying (stages 4 and 5) and after she bought (stage 6). In particular, we want to prevent the possibility of any coalition of merchants in electronic commerce. This could help in the fight against spam as well as against various forms of indiscretion, protecting Alice's privacy.

According to our second objective, we make sure that Bob's catalogue is kept secret, in particular from other sellers who might try to masquerade as buyers. Let us imagine a world in which no seller knows the selling price asked by the competition. How would things go in such a world? Admittedly, there is not an exhaustive list of all negative or positive possibilities that might arise. Nevertheless, three significant elements can be considered. Firstly, each seller should sell at the lowest price that he can afford, taking into account his cost price. Secondly, there would be no unfair competition. And finally, there would be no sociological elements (race, religion, distinction between private and public companies, government, etc.) to consider.

Working in the footsteps of Chaum, we believe that privacy is a *fundamental* right for all humans, and every means to protect it should be given serious consideration. In particular, no individual should ever have to justify a wish for privacy, and such wish should never be considered suspicious *a priori*. After all, Article 12 in the *Universal Declaration of Human Rights* [31] states that "No one shall be subjected to arbitrary interference with his privacy". Should our idealistic position be considered extreme by some, at least it serves to move the middle-ground in the right direction!

3. RELATED WORK AND CRYPTOGRAPHIC PRELIMINARIES

We review the cryptographic primitives needed for our protocols in the BCBB model, which are based on notions such as Secure Two-Party Computation (STPC), Oblivious Transfer (OT), Private Information Retrieval (PIR), CAPTCHAs and Anonymous Surfing. For completeness, we also review the general notion of Public Key Cryptosystems, which is at the heart of our techniques, as well as its most relevant implementation, which is ElGamal's cryptosystem.

3.1 Secure Two-Party Computation (STPC)

Secure Two-Party Computation (STPC) is the problem of evaluating a function $f(x, y)$ for which the first party, Alice, provides the secret input x and the second party, Bob, provides the secret input y, such that the output becomes known to both parties while the inputs x and y remain secret from Bob and Alice, respectively. The first general STPC protocol was given by Yao [42]. By assuming the intractability of factoring, he showed that every two-party interactive computational problem has a private protocol. Kilian [21] showed later that the STPC problem can be reduced to Oblivious Transfer (see below). Although general techniques (for example [42, 43, 21]) for STPC can be used to solve any such problem, we aim to use the specific structure of our e-commerce setting in order to seek more efficient solutions.

3.2 Oblivious Transfer (OT)

The notion of Oblivious Transfer (OT) was originally conceived by Wiesner [41] under the name "multiplexing channel" in a paper written *circa* 1970 but unpublished until 1983. It was reinvented independently under a simpler form by Rabin [33] in 1981 according to the following scenario: Bob (the sender) has one bit in mind and he wants to transmit it to Alice (the receiver), but through a channel that has probability $\frac{1}{2}$ to deliver. Alice will know whether or not she received the bit, but Bob will not know, and neither party can influence the 50% probability of success. After Rabin's paper, variations on OT appeared in the world of cryptology, such as *1-out-of-2 OT*, or OT_1^2, introduced by Even, Goldreich and Lempel [17] in 1985. (In fact, Wiesner's original multiplexing channel resembles EGL's concept of OT_1^2 more than Rabin's earlier OT.) According to OT_1^2, Bob has two bits (b_0, b_1) in mind. Alice wants to get one of the bits b_i at her choice. At the end of the process, Bob should have learned nothing about i, and Alice should have learned nothing about the other bit b_{1-i}. Furthermore, Alice should not be able to get any joint information on both b_0 and b_1, such as their exclusive-OR.

In blissful ignorance of all the previous work, yet another variation on OT was proposed in 1986 by Brassard, Crépeau and Robert [6] under the name ANDOS ("All or Nothing Disclosure of Secrets"), now known as OT_1^n. According to OT_1^n, Bob has n input *strings* $X_1, \ldots, X_n$ and Alice wants to learn one of them, X_i, for some $1 \leq i \leq n$ of her choice. At the end of the process, Bob should not have learned anything about i and Alice should not have learned any joint information about Bob's strings. Later, Crépeau [14] showed that all these flavours of OT are equivalent to one another.

Our work needs an efficient implementation of OT_1^n, such as what has been proposed by Naor and Pinkas [30, 29, 37]. These implementations use the Private Information Retrieval (PIR) protocols that we describe in the next subsection.

If Alice buys a digital product, we wish to prevent Bob from knowing what he has sold. In particular, this requires the price of the product to be hidden from Bob. Following the idea of a "priced oblivious transfer" by Aiello, Ishai and Reingold [1], Alice keeps in an encrypted form the amount of money that she intends to spend in Bob's shop. For each

transaction, Alice cannot buy the product if her balance is insufficient. If Alice succeeds in buying the product, Bob is able to update the encrypted balance without knowing what it is. We mention that the solutions proposed in [1] have to be adapted because they did not take into account our wish for the anonymity of buyers.

3.3 Private Information Retrieval (PIR)

Private Information Retrieval (PIR) schemes were first introduced by Chor, Goldreich, Kushilevitz and Sudan [12]. A PIR scheme allows the user to privately query the i-th bit, x_i, out of an n-bit string, $x = x_1x_2 \cdots x_n \in \{0,1\}^n$, stored at a server, without revealing any information to that server. If i is the index of the bit the user is retrieving, the PIR scheme does *not* guarantee the privacy of the other bits x_j, $j \neq i$. In the PIR setting, the n-bit string x is considered to be the database stored at the server. The main concern of PIR is to minimize the asymptotic complexity of the number of bits sent from the server to the user. The simplest way to achieve PIR is to have the server send the entire database to the user; this is known as the *trivial scheme*. Using the trivial scheme, the communication complexity is obviously n bits.

To achieve sublinear communication complexity, several PIR schemes have been proposed [12, 4, 22, 7, 23], both *information theoretic* and *computational*. The information-theoretic setting was introduced in [12]; it requires that the queries sent by the user give no information whatsoever to the server about i. The following results have been proved in [12]. First, any information-theoretic PIR scheme with a single database *requires* $\Omega(n)$ bits of communication complexity; hence it cannot be made significantly better than the trivial scheme. Second, to reduce the communication complexity to be sublinear, the data must be replicated across several servers, which are trusted not to communicate with one another. In the computational setting, some interesting PIR schemes have been proposed by Chor and Gilboa [11], Ostrovsky and Shoup [32], Kushilevitz and Ostrovsky [22, 23], Cachin, Micali and Stadler [7], Chang [8], Lipmaa [24]. They are based on intractability assumptions such as: quadratic residuosity assumption [22], $\phi-$hiding assumption, trapdoor one-way permutation [23], etc. These schemes use a single server to hold the database and reduce the communication complexity to be sublinear. There are some other PIR settings. The hardware-based PIR [35] assumes a tamper-proof device, known as *secure coprocessor*, that is hosted at the server and is used as a black box to hold, encipher and transmit information requested by the user. There is also the PIR with pre-processing and off-line communication [3, 34]. Using the secure co-processor idea, Asonov and Freytag [2] proposed a PIR scheme with $O(1)$ on-line computation and communication with periodical off-line pre-processing and zero off-line communication.

PIR schemes only guarantee the privacy of the user, and not that of the server's data. Multi-server *Symmetrically* Private Information Retrieval (SPIR) schemes were introduced in the computational [22] and in the information-theoretic [18] settings in order to ensure privacy of the database in PIR schemes. More precisely, SPIR schemes maintain the user's privacy but prevent him from obtaining any information from the server other than one single physical bit. Single-server SPIR schemes have also been proposed in the computational setting [18, 28, 2]. We also mention that it was proved in [25] that SPIR and OT_1^n are equivalent.

3.4 Captcha

One of our objectives is to provide tools that will help Bob keep secret the catalogue of what he has for sale. But of course, he must allow customers to find out whether or not he can provide what they are looking for. Because his services are accessed through the Internet, hostile competitors could design robots to masquerade as potential customers who would enquire about the availability of every possible product that one could imagine, thus plundering Bob's catalogue in the blink of the eye! To prevent this type of attack, we require customers to be computer-assisted *humans*, even though Bob's agent can well be a computer.

CAPTCHAs are programs designed to create tests that other programs cannot pass but that are easy for humans [40]. For example, a CAPTCHA could choose a random English word, distort the letters in some nasty way, and flash the distorted text on Alice's screen. If Alice-the-human is really there, she can easily type out the chosen word, but if Alice is in fact a hostile computer, it will be unable to do the same *even if Alice-the-program knows the process used by the* CAPTCHA *to distort letters*. CAPTCHAs provide exactly the tool needed for Bob's computer to make sure that he is dealing with flesh-and-blood humans.

3.5 Anonymous Surfing

The Internet is based upon the principle of transferring information from one computer to another. Here, the information can be the data collected while visiting web pages, sending or receiving e-mail, using a chat room, etc. For the purpose of information transfer, each computer has an *identity* known as its Internet Protocol (IP) address. The IP address can be dynamic (changing with almost each Internet connection) or static. Most users with fixed Internet connections (e.g. cable modems) have static IP addresses, while those with dial-up Internet connection are usually given IP addresses dynamically each time they connect.

Users can be traced from their IP addresses [38]. In fact, the IP address is *personally identifiable information* that is automatically captured by another computer whenever a communications link is made over the Internet. Therefore, several methods can be used in order to trace a given user.

1. The IP addresses are distributed in blocks to the Internet Service Providers (ISP) and databases of these blocks are publicly known and available for searching.
2. The name of the owner of a given IP address can be found by using the Reverse Address Resolution Protocol (RARP), also called the *Reverse Lookup*.
3. One can conduct a *Traceroute*, which helps to find the way followed by the *packets* from the origin to the final destination. Concretely, the packets travelling through the Internet pass through several computers in a hierarchical order. For example, information packets can pass from the origin computer to its *attached* ISP, until

it reaches the computer's *Backbone* Provider. The information packets then transfer to the destination Backbone Provider down to the ISP of the destination computer and finally to the intended recipient. It is possible that the information packets do not reach the backbone provider if the computer of the intended recipient has its ISP situated in the hierarchy between the origin ISP and the final backbone.

4. One can review domain registration information, for instance by performing a standard Unix/Linux `whois` command on the domain name portion of the web site of interest. For example, "`whois amazon.com`" provides registration information on `www.amazon.com`.

5. It is possible to search for the IP address and/or computer name of a given user on the Internet. New technologies such as Web bugs and Spyware help many web sites to gather information on web users.

We are more concerned by the last point. If the seller (Bob) knows the IP address or computer name of the consumer (Alice), then there is no privacy anymore.

In our context, we want (1) Alice surfing anonymously in Bob's web site and (2) Bob delivering electronically a digital product to Alice without needing to know information on Alice's identity (*real* IP address, name, etc.). We note that condition (1) implies condition (2): if Alice surfs anonymously, then any digital product must be delivered anonymously as well.

So, how should one surf anonymously? It is clear from the above that privacy preserving requires users to hide the IP addresses of their computers while on the Internet. A simple "solution" consists in the use of Internet cafés, but this is obviously not always convenient. The standard use of *proxies* offers some level of security. If a proxy sits between the users and the Internet, then all users appear to come from the same computer. Therefore, users cannot be traced further than the proxy unless additional information is known. An example of additional information can be the name of the big city near the user, usually included in the ISP's computer names. Also, the registration information given by the user to his ISP for identification cannot be assumed to be in a safe place. Generally, the law prohibits that registration information be given (or sold!) to third parties, but there is no realistic way to enforce this. Moreover, if the user identification by IP address is ambiguous (i.e. if it is difficult to trace users beyond the proxy), the use of *Internet Cookies*, *Spywares*, etc. opens a way for web sites to give a unique identity to a user.

To address these privacy concerns, *trusted identity proxies* such as `Guardster.com` [19] were developed starting in 2002. These proxies offer the possibility of surfing on the Internet without giving away personal information to other web sites. This is done by rewriting the user's request via the browser and cleaning the resulting pages that are returned to the user. See Figure 1 (inspired by [19, *About Us*]). The trusted identity proxies help anonymous surfing by blocking cookies and annoying ads, stopping javascript, and hiding the user's identity and IP address. However, this solution assumes

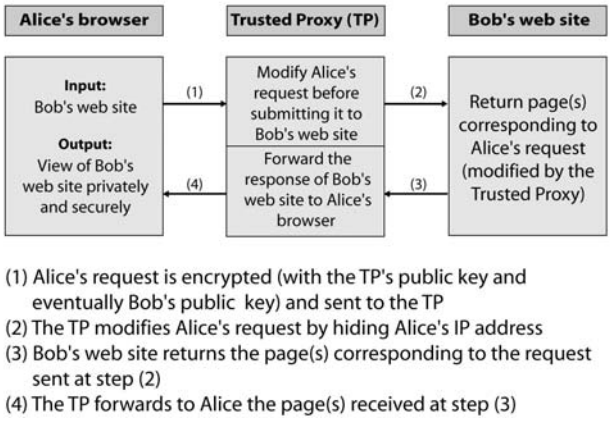

Figure 1: Hiding the IP address

that the identity proxy is indeed trustworthy, and the user's privacy depends crucially on this assumption.

In a seminal 1981 paper, Chaum [9] proposed the technique of *mix-nets*, based on then-emerging Public Key Cryptography (see Section 3.6), to implement "untraceable electronic mail, return addresses and digital pseudonyms". According to this proposal, an electronic mail system can hide the identity of email senders (as well as the content of the communication), yet provide the receiver with the possibility of sending back his response to the right person through an untraceable return address. This approach remains computationally secure even if the underlying telecommunication system is insecure. The advantage of Chaum's technique is that it does not require a *common* Trusted Authority. In fact, each participant has to be considered as an Authority, so that Chaum's solution can be compromised only by subversion or conspiracy of a significant subset of the users. The price to pay for this increased security is that Chaum's approach is less efficient than trusted identity proxies. Also, it remains to be upgraded from its original purpose of email exchange to the more challenging goal of web browsing. (Chaum's paper predates the World Wide Web by one decade!)

We use various forms of anonymous surfing throughout our protocols to protect Alice's identity and to allow her to download digital goods she may decide to buy. It is up to Alice to decide which approach she prefers to use—Internet café, trusted identity proxies or mix-nets—depending on her personal trade-offs between security, efficiency and convenience. (Of course, trusted identity proxies could be used to perform all four steps of our Blind Customer Buying Behaviour model, with no needs for our protocols, but this would require more trust in the proxies than we are willing to grant them, and it would burden them with tasks that they were not set up to accomplish.)

3.6 Public Key Cryptosystems (PKC)

Public Key Cryptosystems (PKCs) were introduced independently by Merkle [27] and by Diffie and Hellman [15]. Formally, a PKC consists of three efficient algorithms:

a *Key-Generation Algorithm* that generates pairs of Secret Key (SK) and Public Key (PK); an *Encryption Algorithm* that computes the ciphertext for a message, given the public key; and a *Decryption Algorithm* that computes the cleartext message back from the ciphertext, given the secret key. PKCs can be *probabilistic*, in which case a *randomization set* R is involved. In addition to the cleartext and public key, the encryption algorithm takes a randomly chosen element of R in order to produce the ciphertext. The decryption algorithm, on the other hand, needs only the ciphertext and the secret key in order to recompute the cleartext. One advantage of probabilistic PKCs is that they prevent a *guessing attack* by which one could verify if a given cleartext is correct by encrypting it with the public key and comparing the resulting ciphertext with the one whose decryption is being seeked.

In e-commerce, the entities that communicate with one another may have no prior relationship between them, so it is important to use PKCs. Often, Alice may wish to buy a product from Bob's online shop even though they have never met before. Alice can just use Bob's public key in order to send him a query. Conversely, Bob sends a response to Alice using her public key. In this context, Bob (resp. Alice) uses his (resp. her) secret key and the decryption algorithm to decipher the query (resp. response). We largely use PKCs to implement the protocols we propose in this paper.

3.7 The Diffie-Hellman Problem (DHP)

Let $G = \langle g \rangle$ be a cyclic group for which the Discrete Logarithm Problem (DLP) is hard, which is the problem, given $a \in G$, of computing x, $0 \leq x < |G|$, such that $a = g^x$ in group G. It is well-known that computing g^x in G requires at most $2 \lg |G|$ group operations using the square-and-multiply algorithm [36], but it is believed that the DLP—the reverse operation—is hard in well-chosen groups.

The Diffie-Hellman Problem (DHP) on G is the problem of computing g^{xy} given g^x and g^y; again, all computations are performed in group G. From a security point of view, the DHP is *at most* as hard as the DLP and it has been shown that the two problems are equivalent for many groups [26].

3.8 ElGamal's Cryptosystem

Given a multiplicative group G and an element $g \in G$, let us consider G', the subgroup of G generated by g. Let n be the order of g in G, and take $\mathbb{Z}_n$ as the randomization set. The ElGamal Cryptosystem [16] is a probabilistic PKC in which the plaintext space is $\mathcal{M} = G$ and the ciphertext space is $\mathcal{C} = G' \times G$.

- Key-Generation Algorithm: choose $d \in \mathbb{Z}_n$ at random. Set SK $= d$ and compute PK $= k = g^d$.
- Encryption Algorithm: For each public key $k \in G'$, the encryption function $E_k : \mathcal{M} \times \mathbb{Z}_n \to \mathcal{C}$ is defined by $E_k(m, r) = (g^r, m \cdot k^r)$.
- Decryption Algorithm: For each public key k, recall that the corresponding secret key is the integer $d \in \mathbb{Z}_n$ such that $k = g^d$. The decryption function $D_d : \mathcal{C} \to \mathcal{M}$ is defined by $D_d(e_1, e_2) = e_2 \cdot (e_1^d)^{-1}$.

In practice, the group G must be chosen so that the DLP is hard. It is known that breaking ElGamal's cryptosystem by a ciphertext-only attack is equivalent to solving the DHP. In the BCBB model, we use ElGamal's cryptosystem to encipher the exchanges between Alice and Bob (see Section 4).

4. BLIND CUSTOMER BUYING BEHAVIOUR (BCBB) MODEL

In this section, we present the Blind Customer Buying Behaviour (BCBB) model. The BCBB model requires four protocols operating on the six stages of the standard CBB model: the BliS protocol (stages 1 to 3 of CBB), the BliN protocol (stage 4), the BliP protocol (stage 5) and the BliM protocol (stage 6). In this paper, we concentrate on the BliS protocol in Section 4.2, and we give sketchy solutions for the BliN, BliP and BliM protocols in Sections 4.3, 4.4 and 4.5. Those three protocols will be the subject of a follow-up paper because we have not yet worked them out in detail. But we must first address the question: What is it that can and should be kept secret?

4.1 Some things are more secret than others

From Alice's point of view, the situation is clear: She wants to keep secret her identity, what she is shopping for, and what she finally decides to buy. All these things are possible if she is shopping for a digital product, which can be downloaded and upgraded through anonymous channels. In case of physical products, the situation is more complicated, requiring anonymous distribution centres.

Ideally, Bob would like to keep his entire catalogue secret for fear of hostile competition. He does not want other merchants to know what he has to sell and at what price. But keeping the catalogue completely secret makes no sense whatsoever! How could you buy from a shop that refuses to tell you whether or not what you are looking for is for sale? For this reason, we distinguish between items that Bob would prefer to hide, but that he is nevertheless willing to reveal at a slow rate (and only to humans), and those that are highly sensitive. For the purpose of this paper, we shall consider that the only highly sensitive item is the price, but this notion can be generalized easily. For example, a negotiation strategy would also be highly sensitive. We shall refer to the less sensitive information as the Data Semi-Privacy (DSP) subset of the catalogue, whereas the highly sensitive information shall be termed its Data Full-Privacy (DFP) subset.

4.2 The Blind Search (BliS) Protocol

We are now ready to present our Blind Search protocol in detail. This protocol requires Alice to surf anonymously in Bob's web site. If unwilling to trust identity proxies, Alice and Bob could make use of a system inspired by Chaum's [9], so that every request sent by Alice is considered as untraceable electronic mail and Bob answers (web pages, delivery, etc.) via Alice's untraceable return address.

BliS protocol overview: We suppose that Bob has t different products to sell, each of them having characteristics chosen among m possibilities. The set $C = \{C_1, \ldots, C_m\}$ of characteristics is publicly known. Bob's database can be represented as a table T of t lines and $m + 2$ columns: one

column for each of the m possible characteristics, one for the "state" and one for the "price tag". The column *state* contains an explicit message for each line of T. This message could be: *I have it in stock*, *I don't sell it any more*, *This product has been replaced by [...]*, *I will be supplied in two weeks*, etc. (The message *I don't have it* is unlikely to appear because it would be more natural in this case not to have the corresponding line in the catalogue at all.)

When Alice wants to check on the availability and price of some item in Bob's catalogue, she decides on which characteristics to query. As explained in the Introduction (under "Problem Statement"), Alice could start with a single characteristic of a general nature, such as asking "Do you sell shirts?", and move on to sending more precise queries as the process goes on. Let $Q \subseteq \{1, 2, \dots m\}$ denote the set of characteristics that are of interest to Alice, meaning that she puts $i \in Q$ whenever she cares about characteristic C_i. She tells Bob explicitly what the set Q is.[1] Bob then chooses the corresponding messages stored in the column "state" and makes some operations on these messages. The number of lines concerned by the query based on the set Q can be different from (but at most equal to) that of T. Thus, Bob rearranges the messages contained in column "state" by considering the *most significant* one: we call that the *Message Replacement Procedure* (MRP). For example, if the contents of two lines restricted to the columns in Q are the same, and Bob has two messages, let say *It's in backorder*, and *I have it*, then only one line will be (blindly) presented to Alice with the latter message (*I have it*). Moreover, if two lines differ only in the price column, then Bob chooses the line with the higher price.

Using the process described below, Alice gets to learn whether or not there is a line in Bob's catalogue that fits her set of characteristics, and if so she gets the corresponding "state". At this point, she may either *refine* her request by adding more characteristics, *update* her request by changing some of her current characteristics (such as changing the colour from "red" to "green"), *change* her request altogether by choosing a new set of characteristics that is neither the same nor a superset of the previous set, decide to *leave* Bob's shop without buying, or *query the price* asked by Bob for that item or an earlier one (after seeing something of interest, Alice may wish to continue browsing, but finally decide to buy the earlier item). The last option is qualitatively different from the others because that is the only one that would leak a Data Full-Privacy (DFP) item—see Section 4.1. Therefore, Bob will allow Alice this option only once in a session. But as long as the price has not been queried, Bob will allow Alice to refine, update and change her query. However, CAPTCHAs will be used to prevent robot attacks on the DSP parts of the catalogue. There will be a CAPTCHA each time Alice changes her query and at regular intervals when she updates it. There is no need for a CAPTCHA when Alice refines her query since this would be too annoying for a human who takes this normal route to converge on the product she wants to buy.

[1] This will leak some information to Bob about Alice's needs. For example, if one of the characteristics that Alice puts in Q corresponds to "colour", Bob will infer that Alice is not interested in buying water. But of course Alice can inject pointless characteristics in her set Q, just to mislead Bob.

BliS protocol setting: Let $\mathcal{D}$ be a *Universal Set* for describing any product. That is, $\mathcal{D}$ is a set of all the characteristics that can be used to describe a product such as its name, colour, width, weight, etc. Each characteristic can take one or several values. For example, the characteristic "colour" can have values *red, blue, yellow*, etc. We propose the creation of a standard and universal *codification* for the values that a characteristic can take. Thus, Alice can describe the product she needs by using this codification. Now, she applies a standard hash function H on the values of the characteristics she chose in her set Q. The result of this hash function is called the *digest* of Alice's request. Bob applies the same hash function on the lines and columns from his catalogue that are selected with respect to Q. We refer to this indexing procedure based on H as the *Universal Indexation* (UI) procedure; the function H itself is a *Universal Indexation function* or *UI function*. For now, Alice's request can be summarized as "Give me the information corresponding to the digest codification that I have". But of course, she will communicate that request to Bob in a blind way, as explained below. As for Bob, he has *effective* indices in his table T of products. But the virtual table T' he created with respect to Q has new indices and he knows the correspondence between these new indices and the effective indices in T. In other words, the UI creates a virtual intermediary between Alice and Bob.

In what follows, the variables t and m, as well as the set of characteristics $Q = \{q_1, q_2, \dots, q_{|Q|}\}$, $|Q| \leq m$, are the same as defined in the BliS protocol overview. After Bob receives the set Q and selects the t_Q lines from his catalogue that correspond to Q, we note by $d_{\ell,j}$, the value contained in line ℓ and column q_j, $1 \leq j \leq |Q|$, of table T'. For each line ℓ generated from Q, Bob uses the UI function H to obtain the digest $H(d_{\ell,1}, \dots, d_{\ell,|Q|})$. If v_i denotes Alice's choice for characteristic C_{q_i} (for instance, C_{q_i} could be "colour" and v_i could be the universal code for "red"), Alice's query is codified by digest $H(v_1, \dots, v_{|Q|})$.

Formal presentation of the BliS protocol:

1. Bob picks at random a prime number p, an integer $x \in \{1, \dots, p-1\}$ and a generator g for the group $\mathbb{Z}_p$. He computes $s = g^x$ and sends p, g and s to Alice, but he keeps x secret. [All computations in this protocol are performed in group $\mathbb{Z}_p$ (i.e. modulo p). We assume that prices can be represented by elements of $\mathbb{Z}_p$.]

2. Bob picks a random element $\wp \in \mathbb{Z}_p$, which he keeps secret for now. A list of the chosen $\wp$'s (we may come back to this step several times) is kept by Bob for future use.

3. Alice chooses a set Q of characteristics that she wishes to query and announces it to Bob. For each $q_i \in Q$, she secretly chooses the codification v_i corresponding to what she wants for characteristic C_{q_i}. She applies the UI function H to compute her digest $u = H(v_1, \dots, v_{|Q|})$. She picks at random $a \in \{1, \dots, p-1\}$, computes her blind query $y = u \cdot g^a$, and sends it to Bob.

4. Using the MRP, Bob forms table T' that corresponds to Alice's set Q. That table contains $t_Q \leq t$ lines

among which Alice is looking for a product. Bob applies the UI function H on each line ℓ of T' and gets the associated digests $e_\ell = H(d_{\ell,1}, \ldots, d_{\ell,|Q|})$, $l = 1, \ldots, t_Q$. For each line ℓ, let m_ℓ be the message contained in the column "state" and w_ℓ be its price tag (codified as an element of $\mathbb{Z}_p$). Bob encrypts the price as $z_\ell = \wp \cdot w_\ell$ and he sends messages $M_\ell = m_\ell || z_\ell$, $l = 1, \ldots, t_Q$, to Alice, but in their blind forms:

$$E_\ell = \left((y/e_\ell)^x\ ,\ g^{b_\ell}\ ,\ M_\ell \cdot (y/e_\ell)^{b_\ell}\right),$$

where b_ℓ is picked randomly in $\{1, \ldots, p-1\}$ for each $l = 1, \ldots, t_Q$. He sends E_ℓ, $l = 1, \ldots, t_Q$, to Alice.

5. Alice has received t_Q triples $(\alpha_\ell, \beta_\ell, \gamma_\ell)$ from Bob, but at most one of these is relevant to her request: the one corresponding to ℓ such that $u = e_\ell$, if it exists. To locate it, notice that $\alpha_\ell = (y/e_\ell)^x = (u \cdot g^a/e_\ell)^x$, which is equal to $g^{ax} = s^a$ if and only if $u = e_\ell$. Thus, Alice computes s^a once and for all, and she screens all the triples received from Bob until and if one comes so that $\alpha_\ell = s^a$. At this point, Alice computes

$$\gamma_\ell \cdot (\beta_\ell^a)^{-1} = M_\ell = m_\ell || z_\ell\ .$$

She examines the message m_ℓ, which tells her about the product's availability. At that point, Alice can do one of three things.

   (a) If Alice wishes to continue browsing Bob's catalogue—either because she is dissatisfied with the current offer or because she hopes to find something even better—she readjusts the set Q and goes back to step 2 in the protocol, possibly after Bob subjects her to a CAPTCHA.

   (b) If Alice is satisfied with her browsing, either because she is interested in the current offer or in some offer from an earlier round, she uses Oblivious Transfer to obtain from Bob the value of $\wp$ that was in use during the round of interest. This allows her to decipher the corresponding price $w_\ell = z_\ell \cdot \wp^{-1}$. At that point, Alice may either tell Bob that she is not interested, decide to buy at that price (thus moving on to the Blind Payment protocol—BliP), or move on to the Blind Negotiation protocol—BliN.

   (c) If Alice decides that she is in the wrong place, she informs Bob that she quits and the entire protocol aborts.

4.3 The Blind Negotiation (BliN) protocol

We recall that in the negotiation step (CBB model), Alice and Bob use strategies in order to make choices and decisions. Clarke [13] defines the *Negotiation* as "a process involving dealings among persons, which are intended to result in an agreement, and commitment to a course of action". In our case, the negotiation between Alice and Bob should be about the terms of the transaction that binds them, which are the price and perhaps some other aspects such as the conditions of sale, warranty, etc.

This is all easier said than done when we remember the meaning of *blind* negotiations! We are asking Bob to negotiate with Alice the price of an item he has for sale, but he must do so without knowing his own starting price, nor even what item is being negotiated. Furthermore, he must never get to learn anything about Alice's offers or the final price they will hopefully agree upon. This may sound like Mission Impossible, and indeed a complete solution goes beyond the scope of this paper.

Simple strategies are based on Yao's *Millionaire's Problem* [42] or on Boudot's "efficient proofs that a committed number lies in an interval" [5]. Recall that Yao's classic problem takes place between two millionaires who want to know who is richest without either one telling the other how much he or she is worth. Mathematically, Alice and Bob have secret numbers x and y in mind, respectively, and they want to determine which is largest while keeping those numbers secret. In a negotiation setting, x could be Alice's best offer and y could be Bob's best price. If they can determine whether or not $x \geq y$, they will at least know if it is worth spending time negotiating. The difficulty here is that Bob cannot run his share of Yao's protocol because he does not know his own best price! To solve this problem, Bob must have an additional column in his catalogue, which gives his best price for each item. This information belongs to Bob's Data Full-Privacy (DFP) subset—see Section 4.1. When Alice decides to ask for the price at the end of the BliS protocol (Section 4.2), she obtains the Price-in-the-Catalogue w_ℓ in the clear, as we have seen, but she also obtains an encrypted version of Bob's best price. This encrypted price allows Bob to play a variation on Yao's Millionaire's Protocol even though neither Alice nor Bob will ever get to know that best price in the clear. Details will be provided in a subsequent paper.

To make blind negotiations even more interesting, Bob could have an encrypted pointer to a sophisticated multi-criteria negotiation strategy in yet another DFP column in his catalogue. Eventual promotions could also enter the negotiation process, such as "*Buy three for the price of two*", "*30 Days FREE trial: Satisfaction or Money Back!*", etc. We have not yet worked out the full generality of these ideas, which are left for further research.

4.4 The Blind Payment (BliP) protocol

The Blind Payment (BliP) protocol is about Alice transferring untraceable money to Bob's account in order to receive the product she is buying. We combine Chaum's idea [10] and the protocol proposed by Aiello, Ishai and Reingold [1] to build the BliP protocol. From [10], we retain that Alice deposits money in Bob's account in an untraceable fashion, using a pseudonym that she will employ later to buy products in Bob's shop. The protocol described in [1] is about the selling of digital products by means of the "Priced Oblivious Transfer". More precisely, after making an initial deposit in Bob's account, Alice can engage in several buying transactions. The protocol of [1] allows Bob to debit Alice's account by the amount of the transaction even though he does not know this amount in the clear; it also allows Bob to make sure that Alice's account does not go negative. Again using ideas from [10], Alice can pump additional money in her account at Bob's whenever that becomes necessary. In this way, she can use that account as long as she wishes to buy products from Bob. Whenever Alice decides to close her account at Bob's, she receives an anonymous electronic cheque from him, which she can deposit under a different

pseudonym in her main bank account or, if she prefers, in an account she could have in a different shop.

The BliP protocol is also concerned with product delivery. In the case of a *digital* product, there are several possibilities. If Alice sits in an Internet café, the product is downloaded directly, preferably on Alice's USB key if the café's policy allows it. If Alice trusts an identity proxy, she can use it to recover the product in the comfort of her home or office. Otherwise, Chaum's concept of untraceable return address can be used. In a follow-up paper, we shall propose a mechanism to facilitate the anonymous delivery of *physical* goods by way of untrusted third parties.

4.5 The Blind Maintenance (BliM) protocol

In the Blind Maintenance (BliM) protocol, we propose a blind update scheme that Alice can use in order to download the last updates or corrective measures available in Bob's database. We suppose that Alice has bought a digital product from Bob. Now, she wants to know if there are some updates available in Bob's after-sale service database. For that, she must have obtained a blind service certificate from Bob during the BliS protocol. That certificate corresponds to yet another DFP column in Bob's catalogue. For maintenance purposes, Bob has to keep the last updates and/or corrective measures for his digital products in a table M. The BliM protocol is similar to BliS, but it uses the certificate to query table M in order to get the corresponding update(s) or corrective measure(s).

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have defined the Blind Customer Buying Behaviour (BCBB) model for e-commerce, which protects Alice's and Bob's privacy when they conduct electronic commerce while keeping the spirit of the standard Customer Buying Behaviour (CBB) model introduced in [20]. In particular, we explained in detail how Alice can search for a product in Bob's electronic shop in a way that protects both Alice's privacy and Bob's sensitive information. The other protocols that would be needed to fully implement our BCBB model were only sketched and details are left for further research. Of particular interest is the question of how to design efficient blind negotiation protocols that would allow Bob arbitrarily sophisticated negotiation strategies that he could implement with Alice even though he does not know what it is that he is negotiating.

We also leave for further research a global version of all these protocols, which would allow an unlimited number of privacy-concerned consumers to conduct blind business with equally privacy-concerned electronic shops round the planet. The obvious problem is for consumers to find shops most appropriate to their needs when the shops keep their catalogues secret!

We are aware that not everybody will embrace our wish for privacy. Some people may approve of the constitution of dossiers on their buying habits because that might increase the probability that they receive occasional relevant spams. Similarly, some shops want to advertise their prices loud and clear because they believe they cannot be undersold. In conclusion, we suggested an alternative to the standard CBB model, but the final choice belongs to each individual.

6. REFERENCES

[1] B. Aiello, Y. Ishai and O. Reingold, "Priced oblivious transfer: How to sell digital goods", *Advances in Cryptology: Proceedings of Eurocrypt 01*, pages 119–135, Springer-Verlag, 2001.

[2] D. Asonov and J.-C. Freytag, "Almost optimal private information retrieval", Technical report HUB-IB-156, Humboldt University, Berlin, 2001.

[3] F. Bao, R. H. Deng and P. Feng, "An efficient and practical scheme for privacy protection in the e-commerce of digital goods", *Proceedings of 13th International Conference on Information Security and Cryptology*, pages 162–170, December 2000.

[4] D. Beaver, J. Feigenbaum, J. Kilian and P. Rogaway, "Locally random reductions: Improvements and applications", *Journal of Cryptology* **10**(1):17–36, 1997.

[5] F. Boudot, "Efficient proofs that a committed number lies in an interval", *Advances in Cryptology: Proceedings of Eurocrypt 00*, pages 431–444, Springer-Verlag, 2000.

[6] G. Brassard, C. Crépeau and J.-M. Robert, "All-or-nothing disclosure of secrets", *Advances in Cryptology: Proceedings of Crypto 86*, pages 234–238, Springer-Verlag, 1987.

[7] C. Cachin, S. Micali and M. Stadler, "Computationally private information retrieval with polylogarithmic communication", *Advances in Cryptology: Proceedings of Eurocrypt 99*, pages 402–414, Springer-Verlag, 1999.

[8] Y.-C. Chang, "Single database private information retrieval with logarithmic communication", Available at URL: `http://eprint.iacr.org/2004/036/`, 2004.

[9] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms", *Communications of the ACM* **24**(2):84–90, February 1981.

[10] D. Chaum, "Security without identification: Transaction systems to make Big Brother obsolete", *Communications of the ACM* **28**(10):1030–1044, October 1985.

[11] B. Chor and N. Gilboa, "Computational private information retrieval", *Proceedings of 29th Annual ACM Symposium on the Theory of Computing*, pages 304–313, 1997.

[12] B. Chor, O. Goldreich, E. Kushilevitz and M. Sudan, "Private information retrieval", *Proceedings of 36th Annual IEEE Symposium on Foundations of Computer Science*, pages 41–51, 1995.

[13] R. Clarke, Fundamentals of negotiation. URL: `www.anu.edu.au/people/Roger.Clarke/SOS/FundasNeg.html`, 1993, accessed 8 September 2004.

[14] C. Crépeau, "Equivalence between two flavours of oblivious transfers", *Advances in Cryptology: Proceedings of Crypto 87*, pages 350–354, Springer-Verlag, 1988.

[15] W. Diffie and M. E. Hellman, "New directions in cryptography", *IEEE Transactions on Information Theory* **22**(6):644–654, October 1976.

[16] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", *IEEE Transactions on Information Theory* **31**:469–472, 1985.

[17] S. Even, O. Goldreich and A. Lempel, "A randomized protocol for signing contracts", *Communications of the ACM* **28**:637–647, 1985.

[18] Y. Gertner, Y. Ishai, E. Kushilevitz and T. Malkin, "Protecting data privacy in private information retrieval schemes", *Proceedings of 30th Annual ACM Symposium on the Theory of Computing*, pages 151–160, 1998.

[19] Guardster, Ltd., "Anonymous web surfing", URL: `www.guardster.com`, accessed 8 September 2004.

[20] R. H. Guttman, A. G. Moukas and P. Maes, "Agent-mediated electronic commerce: A survey", *Knowledge Engineering Review Journal* **13**(3):985–1003, June 1998.

[21] J. Kilian, "Founding cryptography on oblivious transfer", *Proceedings of 20th Annual Symposium on Theory of Computing*, pages 20–31, 1988.

[22] E. Kushilevitz and R. Ostrovsky, "Replication is not needed: Single database, computationally-private information retrieval", *Proceedings of 38th Annual IEEE Symposium on Foundations of Computer Science*, pages 364–373, 1997.

[23] E. Kushilevitz and R. Ostrovsky, "One-way trapdoor permutations are sufficient for single-server private information retrieval", Technical report CS0962, Computer Science Department, Technion, 1999.

[24] H. Lipmaa, "Computationally private information retrieval with quasilogarithmic total communication", Available at URL: `http://eprint.iacr.org/2004/063/`, 2004.

[25] T. G. Malkin, "A study of secure database access and general two-party computation", PhD thesis, Massachusetts Institute of Technology, 2000.

[26] U. M. Maurer and S. Wolf, "The relationship between breaking the Diffie-Hellman protocol and computing discrete logarithms", *SIAM Journal of Computing* **28**(5):1689–1721, 1999.

[27] R. C. Merkle, "Secure communications over insecure channels", *Communications of the ACM* **21**:294–299, 1978.

[28] M. Naor and B. Pinkas, "Oblivious transfer and polynomial evaluation", *Proceedings of 31st Annual ACM Symposium on the Theory of Computing*, pages 294–303, 1997.

[29] M. Naor and B. Pinkas, "Oblivious transfer with adaptive queries", *Advances in Cryptology: Proceedings of Crypto 99*, pages 573–590, Springer-Verlag, 1999.

[30] M. Naor and B. Pinkas, "Efficient oblivious transfer protocols", *Proceedings of 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 448–457, January 2001.

[31] Office of the High Commissioner for Human Rights, "Universal Declaration of Human Rights", URL: `http://www.unhchr.ch/udhr/lang/eng.htm`, accessed 8 September 2004.

[32] R. Ostrovsky and V. Shoup, "Private information storage", *Proceedings of 29th Annual ACM Symposium on the Theory of Computing*, pages 294–303, 1999.

[33] M. Rabin, "How to exchange secrets by oblivious transfer", Technical memo TR-81, Aiken Computation Laboratory, Harvard University, 1981.

[34] C. P. Schnorr and M. Jakobsson, "Security of signed ElGamal encryption", *Advances in Cryptology: Proceedings of Asiacrypt 00*, pages 73–89, Springer-Verlag, December 2000.

[35] S. W. Smith and D. Safford, "Practical server privacy with secure coprocessors", *IBM Systems Journal* **40**(3):683–695, September 2001.

[36] D. R. Stinson, *Cryptography: Theory and Practice*, CRC Press, 1995.

[37] C. Tobias, "Practical oblivious transfer protocols", *Proceedings of 5th International Workshop on Information Hiding*, pages 415–426, Springer-Verlag, 2002.

[38] Network-Tools.com, Tools to trace a user, URL: `www.Network-Tools.com`, accessed 8 September 2004.

[39] E. Turban, E. Mclean, J. Wetherbe, N. Bolloju and R. Davison, *Electronic Commerce, A Managerial Perspective*, Prentice Hall, 2002.

[40] L. von Ahn, M. Blum, N. J. Hopper and J. Langford, "CAPTCHA: Telling humans and computers apart", *Advances in Cryptology: Proceedings of Eurocrypt 03*, pages 294–311, Springer-Verlag, 2003.

[41] S. Wiesner, "Conjugate coding", Manuscript written *circa* 1970; Unpublished until it appeared in *Sigact News* **15**(1):78–88, 1983.

[42] A. C.-C. Yao, "Protocols for secure computation", *Proceedings of 23rd IEEE Symposium on Foundations of Computer Science*, pages 160–164, 1982.

[43] A. C.-C. Yao, "How to generate and exchange secrets", *Proceedings of 27th IEEE Symposium on Foundations of Computer Science*, pages 162–167, 1986.