Oblivious transfer protecting secrecy

An implementation for oblivious transfer protecting secrecy almost unconditionally and a bitcommitment based on factoring protecting secrecy unconditionally.

Bert den Boer

Philips Crypto B.V.
Postbus 218
5600 MD Eindhoven
The Netherlands

Abstract

We present a scheme for oblivious transfer in which contrary to earlier proposals the secrecy is protected (almost) unconditional and not by a cryptographic assumption. We also present a bitcommitment scheme based on factoring where the secrecy is unconditional.

I.B. Damgard (Ed.): Advances in Cryptology - EUROCRYPT '90, LNCS 473, pp. 31-45, 1991.

Introduction

An Oblivious Transfer protocol as introduced in [Ra] and with one of the first mathematically based implementations in [Bl] is roughly a protocol where a sender puts a message in a communication channel where the message has a chance of (usually) one half to arrive at the intended recipient. The protocol has two important aspects; one is secrecy: the recipient cannot by deviating from the protocol increase the chance that eventually he gets hold of the message. The other aspect is authentication : the sender cannot find out whether the message reached the recipient. Based on heuristic arguments it is generally believed that both aspects cannot be met unconditionally in a mathematically-oriented protocol. So we need a cryptographic assumption to protect at least one of the two aspects. In the literature implementations for oblivious transfer based on a cryptographic assumption protected authenticity unconditionally, i.e. the sender could even with infinite computing power never find out what message reached the recipient.

In those protocols the cryptographic assumption is that factoring is a hard problem. So in those implementations we have to believe that the recipient cannot find a particular square root otherwise the aspect of secrecy is not fulfilled.

In our proposal we reverse which aspect is protected unconditionally and which aspect is protected by a cryptographic assumption. In our proposal the cryptographic assumption is the Quadratic Residuosity Assumption and this as-

sumption must protect the authenticity. Our proposal protects secrecy almost unconditionally, i.e. the recipient could even with infinite computer power not find out what the message was if he could not find out what that message was by following the protocol; the word almost reflects the fact that the recipient could cheat in an initializing protocol, but the chance that this goes undetected can be made extremely small and of course if the sender detects this cheating he refuses to do the main protocol.

We also present a bitcommitment scheme. Such a scheme can be derived from our proposal for Oblivious Transfer but our bitcommitment scheme is with completely unconditional secrecy. In the presented form the underlying cryptographic assumption is the factoring problem. An earlier proposal for bitcommitment based on factoring is more efficient but guarantees secrecy only almost unconditionally while our bitcommitment guarantees secrecy unconditionally.

Oblivious Transfer, three flavors

Roughly speaking oblivious transfer is a protocol where a message usually has a chance of $1/2$ of reaching the recipient Bob and where the sender Alice can not find out whether this message reached Bob. For a single message this is a adequate definition. Further we have One out of Two Messages Oblivious Transfer and One Chosen out of Two Messages Oblivious Transfer.

In the first case Alice has two messages, one message will reach Bob but Alice does not know which one and Bob will know that he got say the second message but can not a priori decide which of the two messages he will get. In the second case Bob also gets only one message but this time he is able to decide which of the two he will get while Alice cannot find out his choice.

Description of the Protocol without mathematics

Alice the sender has a one bit message and encodes this message with a coin, say heads up for one. Before she comes to that she takes two coins and two identical wrapping papers and gives them to Bob sitting at the other end of the table. Bob wraps both coins in paper and gives them back to Alice. She unwraps one of the coins and puts the wrapping paper in a low tray. Secondly she puts clearly the still wrapped coin in the tray. And finally she puts the unwrapped coin with the side according to her message bit above ,her choice protected from Bobs sight, in the tray. The tray is filled with a liquid completely dissolving the papers. Alice puts the lid on the tray, shifts the tray long enough for both the dissolving process and the shuffling of the coins. The tray is that low that the coins cannot flip. Then the tray is handed to Bob and he opens the tray and exams the coins, this time such that Alice cannot see which sides are up. If Bob sees two heads he knows that Alice's bit was one but if he sees one head and one tail he does not know which of the two was Alice's choice and which was still wrapped when Alice put it in the tray.

Need for an Initializing Protocol

Basically our protocol works with two coins for a one bit message. Alice chooses the upper side of one coin according to her message and the other is

thrown such that the Alice, the sender cannot see which side is up and such that Bob, the recipient, sees both coins but cannot see which was thrown and which was chosen. We approach this situation under the Quadratic Residuosity Assumption. Given a number N which is not a square and which among its set of divisors has exactly two prime numbers we know from elementary number theory that the group of residues with Jacoby symbol one is twice as big as the subgroup of square residues. Given two arbitrary residues J and *Res* with Jacoby symbol plus one Alice start with encoding her message bit by a square if it is zero and and by J times a square if her message bit is one. This is her first residue. She throws a coin by flipping a coin first and decoding zero by *Res* and one by *Res* times J. This is her second residue. If Bob made N and choosed J and *Res* then Alice does not know whether the second residue is a square or not. So for Alice the second residue is like throwing a coin. Bob cannot influence this coin unless J is a square. But then he cannot decrypt the first residue. So Bob better chooses J to be a non square. Up to now Bob can easily distinguish the (first) residue Alice made according to her choice and the (second) residue made as an implementation of a thrown coin even if Alice arbitrarily interchanges their order or not. The final tuning for the main algorithm is to divide the first residue on the second and sending this quotient residue and the first residue in arbitrary order to Bob. Bob checks that their product is one of the two admitted cases and decrypt a residue into a one if and only if it is not a square. If and

only if both bits are the same he knows the message. This two-steps decoding only works if J is not a square. It is easy to analyze that if J is a square Alice message bit remains unknown.

So the only way to cheat for Bob is making N a square with two prime divisors or make a modulus with three or more prime divisors. The first trick is easy to recognize but to avoid the second way of cheating Bob needs to convince Alice that N has only two prime divisors.

As far as the author knows there is no direct proof for this (in the litterature there is even a claim that deciding whether a number has two or three prime divisors is a hard problem) without giving away the factorization and thereby the authenticity of the main protocol and so we need a interactive protocol(or a mutually trusted random source like in [GP] or [SP]) for convincing Alice that the number N chosen by Bob has only two prime factors.

Initializing Protocol

Bob ,the recipient in the main protocol, produces a R.S.A. modulus N and a non-square residue J with Jacobi-symbol 1. He sends them to Alice, the sender in the main protocol and Alice and Bob participate in a zero-knowledge protocol like the protocol described in [GP] (in their protocol there is no interaction because there the existence of a mutually trusted random source is assumed) where Alice challenges Bob to prove that the number of prime factors of N is equal to

two and J is a non-square. Alice does not need a protocol to find out that N has at least two different prime factors because we assume that the order of the multiplicative group modulo a good R.S.A. modulus has only a factor two or perhaps another small factor in common with $N-1$. The other claimed properties are not feasible to check for Alice on her own so she needs this protocol. We will briefly sketch one round of this protocol. Both produce a residue with Jacobi-symbol 1, first Alice sends her residue to Bob and then Bob sends his to Alice, Alice makes a choice challenging Bob to write either his residue or the product of her residue and his residue as $J^b R^2$. Bob sends the bit b and the residue R to Alice and Alice checks for correctness.

If the number of different prime factors is three or more Bob has a chance of $3/4$ of being able to produce a adequate pair b and R. In the case that N contains only two different prime factors but J is a square Bob also has a chance of $3/4$ of being able to produce b and R. By using n rounds of this protocol with new choices for residues the chance of successfully cheating by Bob becomes $(3/4)^n$

In other words at the expense of linear cost Alice can get a "exponential confidence" in Bob's claim about N and J.

Main Protocol

The first protocol is for Single Message Oblivious Transfer.

Before Alice starts sending messages, she gets a set of residues, called Res, with Jacobi-symbol 1 from Bob. A message is written in binary form and let us say c is the next bit to send. To send this bit of the message Alice produces a random residue S and a random bit d and computes $A = J^c S^2$ and $B = Res\, J^d / A$ where Res is the lowest numbered element of the set of residues Alice got from Bob which is not used for other messages. Then A and B are interchanged precisely if their values were not in a natural order. After this the two residues are send in their natural ordering to Bob. Bob checks that the product is either an element of the list of residues he send Alice at the start of the main protocol or of the form J times such an element (and identical to products of pairs for sending other bits of the same message).

With high probability we may assume that N and J are what they are supposed to be and then Bob will find out what the message bit c is precisely when the product $Res\, J^d$ of the two residues A and B is a square. The chance that this occurs is exactly $1/2$ under Q.R.A.. This chance has this value because Alice is under Q.R.A. not supposed to see the quadratic characteristic of Res and by choosing d independently with a chance $1/2$ from her point of view there is a chance $1/2$ that the product $Res\, J^d$ of the two residues is a square. The reason that Bob can "encrypt" c in the case that the product of the residues is a square is first : Bob can easily see that the product is a square because he knows the factorisation of N. And the second reason for Bob's ability of encrypting is that in that case both residues he got from Alice have to be or both a square or both a non-square. In that case, so although Bob does not know which of the two resi-

dues is the message A containing the message bit c ,he certainly knows whether A is a square (because both residues are squares) or not and therefore what the message bit c is. In case the product is a non square and of the form $Res\ J^d$ (this form disables Alice to know whether or not A and B have the same quadratic characteristic) A and B have a different quadratic characteristic and as Bob can not tell which of the two residues is A , from his point of view it is equally likely that c equals zero as well equals one. This dilemma means that the message does not reach Bob. In this case Bob can study A and B for years but he will never know (without the cooperation of Alice) so if according to the protocol Bob did not receive the message he will not find it by any deviation of the protocol. So Bob can, after committing to a good modulus and non-square residue, never find out the message bit if the product of the two residues is a non-square. So indeed secrecy is almost unconditionally protected.

On the other hand if Alice can distinguish between squares and non-squares (thus violating Q.R.A. in other words breaking the cryptographic assumption) she can for example always sends a message which never arrives. She does this by observing the quadratic characteristic of Res and choosing d zero if Res is a non-square and choosing d one if Res is a square. But as long as Alice does not know the quadratic characteristic of Res she does not know whether she "doubles" her message bit or sends a non-descriptive (unordered!) pair zero and one to Bob. So indeed the authenticity is protected by the cryptographic assumption.

The extra factor J which Alice can use (she can take $Res\ J$ or Res as product

for the residues A and B) is to protect her against an attempt to cheat by Bob by making all *Res* a square.

For each message another element *Res* of the list is used otherwise authentication in a higher sense is violated, i.e., for example assume that for two message bits the probably different pairs A and B have the same product *Res* then Alice does not know whether the first message reaches Bob but she knows that either both message reach Bob or none of the two message bits. This dependency is exploited by using the same product *Res* or *Res J* for sending the bits of a single message and also in the One out of Two Messages Oblivious transfer.

The protocols for One out of Two Messages Oblivious Transfer are done by using the same residue *Res* twice but once with *Res* as product of two residues and once *Res J* as product of two residues. Bob can choose which message he gets if Alice has to use *Res* as the first product and *Res J* as the second product. If Alice is allowed to interchange the products, i.e., she is free to choose *Res* respectively as well as *Res J* respectively for the product of the first pair A and B but then she is obliged to make the second B such that the product of the second pair A and B is *Res J*, respectively *Res*. In this case there is no choice for Bob. Exactly one of the products will be a square and the secret bit put in one of the factors of that product will be revealed. It is also clear that the other product is a non-square, therefore one factor is a square and the other is a non-square. It is equally likely that the square or the non-square contains the message bit so this secret bit will never be known by Bob.

Efficiency Considerations

We can decrease message-lengths in the Single Message protocol by sending only the smallest of the pair A, B and the bit d and if necessary a pointer to *Res*.

We can further avoid modulo divisions by Alice the sender if Bob, who is the recipient and the maker of the modulus, is willing to compute inverses of J and his residues *Res* and send those numbers in corresponding pairs to Alice.

There is also a way to avoid the sending of the batch with the residues *Res* (or one residue *Res* per oblivious transferred message) by letting Alice use outputs of one-way functions. This last is not possible with the One Chosen out of two Oblivious Transfer because there Bob has to ensure that the product of the first two residues A and B is a square and therefore the product of the last two residues A and B is a non-square if he wants to know the first secret bit.

After the next chapter we will indicate how an initializing protocol for our Oblivious Transfer can be done in two and a half round.

Unconditionally secure bitcommitment

Under this heading we present a bitcommitment scheme for which the secrecy of the encrypted bit is unconditionally that is each well-formed encryption of zero is also a possible encryption of one and vice-versa. Before we present this we first make the association between the oblivious transfer above and the bitcommitment presented at the end of this section. In [Cr] a method is presented to derive a bit commitment scheme out of oblivious transfer, especially out of One out of Two Messages Oblivious Transfer. For the general method one has

to do several transfers to commit to one bit. By using one of the One out of Two messages Oblivious Transfer presented above it is clear that the Exclusive-Or sum of the two secret bits is almost unconditionally secret for the recipient Bob. In this case one (instead of several as in [Cr] for the general case) One out of Two messages Oblivious Transfer is enough to commit to a bit.

To clarify this let *Res* be a residue made by Bob. Alice want to commit to bit a. She produces a random bit c and two random residues R and S. She computes the bit $b=c \oplus a$ and $A = J^b R^2$, $B = Res/A$, $C = J^c S^2$, $D = Res\, J/C$. She interchanges the first pair if their values are not in their natural ordering and does the same with the second pair. In a straightforward version this fourtuple is a commitment to the bit a. In fact by analyzing this and thereby reducing the message length we get an earlier proposal for a bitcommitment scheme [BCC] based on factoring which is almost unconditionally secure with respect to secrecy. To reduce the message length Bob and Alice extend the initializing protocol by a zero knowledge proof from Bob to Alice that this special *Res* is a square [like in BCC]. The residues A and B are no longer necessary so the bit c is made equal to the secret bit and only the smallest of the residues C and D (the third element of the special fourtupel) is given to Bob. Alice can "open this bitcommitment" (terminology of [BCC]) in two ways if she can compute a square root of *Res*. So suddenly the authenticity of the bitcommitment is protected by the difficulty of computing this square root (whereas the authenticity of the oblivious transfer generating this bitcommitment was protected by Q.R.A.).

Now we will present our bitcommitment scheme. Like the bitcommitment

scheme of [BCC] it is based on factoring. The secrecy of the encrypted bits in the [BCC] scheme is almost unconditional (it depends like our oblivious tranfer on luck in an initializing protocol). We change this protocol at the sacrifice of more computations to get unconditional secrecy of the encrypted bits. We get complete unconditionality for the secrecy of the committed bit by using repeated squaring.

In order to commit to a bit Alice asks Bob to produce an odd number N between M, a fixed power of two, and $2M$ and a residue K with Jacobi-symbol -1. Bob computes $L = K^M \bmod N$ and sends N, K, L to Alice. First Alice checks this triple. In order to commit to a bit c Alice takes a random residue S and computes $T = L^c S^{2M}$ and sends T to Bob. Bob, who better knows the factorization of N, checks that T has indeed odd multiplicative order. Bob can open T in two ways and therefore the secrecy is unconditional and Alice cannot find an alternative opening of T unless she can factor N. (The power M is taken to ensure that the gcd of $2M$ with the multiplicative order of the group is at most M and of course in practice this gcd is much lower.)

Because every residue which can be written as a a residue in the power M also can be written as a residue in the power $2M$ Bob gets no information about the secret bit c. But Alice can only change her commitment if she can write L as a residue in the power $2M$ and that is equivalent to factoring N.

So here we have a bitcommitment scheme with unconditional secrecy and authenticity based on the difficulty of factoring.

Initializing protocol in a bounded number of rounds

In [SP] a One out of Two Oblivious Transfer is presented where the initializing protocol is a single message (half a round) under the assumption of Trusted Randomness. Their protocol differs in the sense that even if N has two prime factors the recipient can cheat if their y is a square. Such a cheating is conceivable if the Public Random bits are known on beforehand and a recipient with unbounded computing power can thereafter select the prime divisors for his public modulus N. Without the assumption of mutually trusted random bits we can make our initializing protocol in a bounded number of rounds using the footsteps of [BCY] as follows:

Bob sends his modulus.

Alice makes random residues with Jacoby symbol one and raises them to the power 2^M (where N has $M+1$ bits) and sends those outputs to Bob.

Bob makes and sends random residues with Jacoby symbol one.

Alice opens the roots of the residues she send.

Bob computes the products of one residue of his own and one root of Alice (following the ordering) and shows that those products can be divided in two sets, where in each set the quotient of two members is a square.

This is a protocol with two and a half round which need an extra polishing if $N-1$ has Jacoby symbol one and Bob is not willing to disclose whether this number is a square or not. In that case Alice also has for each root to commit to whether twice this root is bigger or smaller than the modulus.

Conclusions

We have made an implementation for three flavors of Oblivious Transfer build from nothing in common by the participants.

References

[BCC] Brassard, G.,Chaum, D. and Crepeau, C., "Minimum disclosure proofs of knowledge", Journal of Computer and System Sciences,vol. 37, no. 2, October 1988, pp. 188-195.

[BCY] Brassard, G.,Crepeau, C.,Yung, M., "Everything in NP can be argued in a bounded number of rounds", Abstracts of Eurocrypt 89.

[Bl] Blum, M., "Three applications of the oblivious transfer: Part I: Coin flipping by telephone; Part II: How to exchange secrets; Part III: How to send certified electronic mail", Department of EECS, University of California, Berkeley, CA, 1981.

[Cr] Crepeau, C., "Verifiable all or nothing disclosure of secrets", Abstracts of Eurocrypt 89.

[GP] Graaf, J. de, Peralta, R., "A simple and secure way to show the validity of your public key", Proceedings Crypto 87, pp. 87-119.

[Ra] Rabin, M., "how to exchange secrets by oblivious transfer", Tech. Memo TR-81, Aiken Computation Laboratory, Harvard University, 1981.

[SP] Santis, A. de, Persiano, G., "Public-Randomness in Public-Key Cryptography", Abstracts of Eurocrypt 90.