Unconditionally Secure Commitment and Oblivious Transfer Schemes Using Private Channels and a Trusted Initializer

Ronald L. Rivest
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139
rivest@mit.edu

November 8, 1999

Abstract

We present a new and very simple commitment scheme that does not depend on any assumptions about computational complexity; the Sender and Receiver may both be computationally unbounded. Instead, the scheme utilizes a "trusted initializer" who participates only in an initial setup phase. The scheme also utilizes private channels between each pair of parties. The Sender is able to easily commit to a large value; the scheme is not just a "bit-commitment" scheme.

We also observe that 1-out-of-n oblivious transfer is easily handled in the same model, using a simple OT protocol due to Bennett et al.[2].

Keywords: bit-commitment, commitment scheme, unconditional security, trusted third party, trusted initializer, oblivious transfer.

1 Introduction

A *commitment scheme* must specify a COMMIT protocol and a REVEAL protocol, involving a Sender (Alice) and a Receiver (Bob). Alice has a secret value x_0 that she commits to in the COMMIT protocol, although Bob learns nothing about x_0 then. Later, Alice discloses x_0 in the REVEAL protocol. Bob may reject the value Alice reveals if it is inconsistent with the information he received during COMMIT: Alice may not "change her mind" about the value to be revealed, once COMMIT is finished. Thus, a commitment scheme must satisfy the following requirements:

Correctness: If both parties are honest and follow the protocols, then during the REVEAL protocol Bob will learn the value x_0 that Alice wished to commit to.

Privacy: Bob learns nothing about x_0 during the COMMIT protocol.

Binding: After the COMMIT protocol has finished, there is only one value of x_0 that Bob will accept during the REVEAL protocol.

If x_0 is required to be a single bit, we say that the commitment scheme is a "bit commitment scheme," otherwise we use the more general term "commitment scheme."

There are many applications for commitment schemes. Sealed-bid auctions are one obvious example: x_0 represents Alice's bid. Commitment schemes are useful for identification schemes[16], multiparty protocols[17], and are an an essential component of many zero-knowledge proof schemes [18, 5, 11].

2 Previous Work on Commitment Schemes

Since commitment schemes were first introduced by Blum[3] in 1982 for the problem of "coin flipping by telephone," commitment schemes have been an active area of research.

However, one must face the "facts of life":

> *"It is well known (and easy to see) that in a two-player scenario with only noiseless communication, OT [Oblivous Transfer] and BC [Bit Commitment] with information-theoretic security is not possible, even if only passive cheating is assuming, and players are allowed infinite computing power."* [12, page 61]

Therefore, researchers have focussed primarily on

- commitment schemes based on computational assumptions, where binding or privacy depends on the Sender or Receiver, respectively, being computationally bounded, or
- commitment schemes based on other models of communication involving noisy channels or quantum mechanics.

If the scheme requires that the Sender be computationally bounded in order to achieve binding, we say that it is *computationally binding*; otherwise it is *unconditionally binding* (some authors would call this *information-theoretically* binding).

Similarly, if the scheme requires that the Receiver be computational bounded in order to achieve privacy, we say that it is *computationally private*; otherwise it is *unconditionally private* (some authors would call this *information-theoretically private*, or *unconditionally hiding*, or *unconditionally concealing*).

It is typically enough to assume that only one of the Sender or Receiver be computationally bounded, so that one achieves an asymmetric result: either computational binding and unconditional privacy, or unconditional binding and computational privacy.

Commitment schemes that are computationally binding and unconditionally private have been proposed by many researchers, including Blum[3], Goldwasser, Micali, and Rivest (implicit in their signature scheme[19]), Brassard, Chaum, and Crèpeau[5], Brassard, Crèpeau, and Yung[8], Halevi and Micali[21], and Halevi[20]. Brassard and Yung[7] develop a very general framework and theory for all bit commitment schemes having unconditional privacy, based on "one-way group actions." Damgård, Pedersen, and Pfitzmann[13] show that the existence of "statistically hiding" bit commitment schemes (which provide nearly perfect unconditional privacy) is equivalent to the existence of fail-stop signature schemes.

Naor[25] presents a commitment scheme that is unconditionally binding and computationally private, based on any pseudorandom generator (or equivalently, based on any one-way function). Ohta, Okamoto, and Fujioka[26] show that Naor's scheme is secure against divertibility, and note that the non-malleable bit-commitment scheme of Dolev, Dwork, and Naor[14] can also be used to provide such protection. Ostrovsky, Venkatesan, and Yung[27] examine in some detail bit commitment schemes when at least one of the Sender/Receiver is computationally unbounded, and in particular show that when the Sender is computationally unbounded, a commitment scheme may be based on any hard-on-average problem in PSPACE.

Some researchers have explored information-theoretic models, based on the assumption of noisy communication channels. For example, Crèpeau[10] improves on his earlier work with Kilian[9] by giving efficient algorithms for bit commitment and oblivious transfer over a binary symmetric channel. Later, Damgård, Kilian, and Salvail[12] explore such questions further based on "unfair noisy channels" and related assumptions.

Other researchers have explored bit commitment in models of quan-

tum computation. Brassard et al.[6] proposed a quantum bit commitment scheme, but a subtle flaw was discovered; Mayers[24] proved general quantum bit commitment to be impossible, as did Lo and Chau[23]. More recently, Salvail[30] shows that under certain restricted assumptions about the Sender's ability to make measurements, quantum bit commitment is still possible.

Bit commitment schemes occur within a wide variety of models and applications, not all of which are mentioned above, or which fit in the above taxonomy. Just to pick one interesting example, Ben-Or Goldwasser, Kilian, and Wigderson utilize a bit commitment scheme in a "multi-prover" model[1]: of two provers who can't communicate with each other, one commits a bit to a verifier, and the other reveals it.

3 Our Model

We believe it is of interest to look for practical commitment schemes that work when both the Sender and the Receiver are computationally unbounded. What would one use for a commitment scheme, for example, if it turns out that $P = NP$?

The normal model (two-party protocol with noiseless channel) does not admit a solution. How can one most simply extend the model to permit a solution?

In this paper we make the following assumptions, which suffice:

- There is a "trusted third party" (Ted). Ted is honest, and both Alice and Bob trust that Ted will execute his role correctly.
- There are "private channels" between each pair of parties; Alice and Bob can each communicate privately with Ted and with each other.

We desire that, if possible, Ted should never find out the value of x_0. Alice and Bob trust Ted to be honest, but the value of x_0 is none of Ted's business, and Alice and Bob prefer that he never learns x_0. This requirement rules out the obvious solution wherein Alice gives x_0 to Ted during COMMIT, and Ted gives x_0 to Bob during REVEAL.

We further desire that, if possible, Ted should not participate in the COMMIT and REVEAL protocols. We prefer a solution wherein Ted only participates in an initial SETUP protocol. In such a scheme, Ted is done before Alice may even have decided which x_0 to commit to. This rules out the obvious solution wherein Ted gives Alice a random string r during

SETUP, Alice gives Bob $r \oplus x_0$ during COMMIT, and Ted gives Bob r during REVEAL.

We call a trusted third party who participates only in a setup phase, before the other parties may even have their inputs, a *trusted initializer*. Protocols using trusted initializers are much easier to implement than more typical protocols using trusted third parties, since the initialization can be performed well in advance of the actual protocol, and the trusted party does not need to be available to participate in the actual heart of the protocol.

4 Our commitment scheme

Our commitment scheme is illustrated in Figure 1.

All computations are performed modulo p, for some fixed suitably large globally known prime number p. We assume that Alice's secret value x_0 satisfies $0 \leq x_0 < p$.

The communications patterns are very simple: during SETUP Ted sends some (different) information to Alice and Bob. During COMMIT Alice sends one number to Bob. During REVEAL Alice sends three numbers to Bob. Each protocol is thus minimal: just one pass.

For the SETUP phase, Ted randomly chooses two numbers $a \in_R \mathbf{Z}_p^*$ and $b \in_R \mathbf{Z}_p$. These numbers define a line:

$$y = ax + b \pmod{p} . \tag{1}$$

Ted sends the values a and b privately to Alice. Ted also picks another value x_1 uniformly at random from $\mathbf{Z}_p$, and computes the value

$$y_1 = ax_1 + b \pmod{p} . \tag{2}$$

Ted privately sends Bob the pair (x_1, y_1); this is a point on the line.

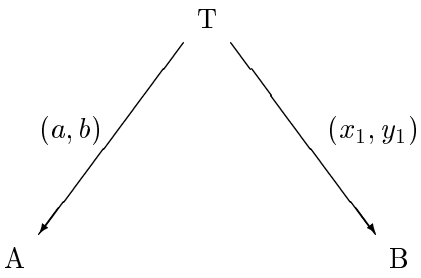
For the COMMIT phase, Alice computes the value

$$y_0 = ax_0 + b \pmod{p} \tag{3}$$

and privately sends the value y_0 to Bob.

For the REVEAL phase, Alice privately sends Bob her secret value x_0, and also the pair (a, b). Bob checks that (x_0, y_0) and (x_1, y_1) satisfy equations (3) and (2). If so, he accepts x_0, otherwise he rejects.

SETUP:



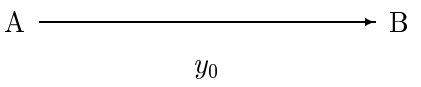
$a \in_R \mathbf{Z}_p^*$

$b \in_R \mathbf{Z}_p$

$x_1 \in_R \mathbf{Z}_p$

$y_1 = ax_1 + b \pmod{p}$

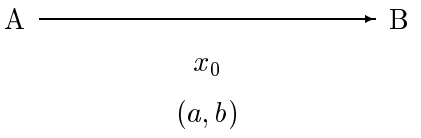
COMMIT:



x_0 is Alice's secret

$y_0 = ax_0 + b \pmod{p}$

REVEAL:



Bob checks that

$y_0 = ax_0 + b \pmod{p}$

and that

$y_1 = ax_1 + b \pmod{p}$.

Figure 1: Our commitment scheme: Alice commits the value x_0 to Bob, using the assistance of a trusted initializer, Ted.

5 Analysis

Theorem 1 *The proposed commitment scheme is unconditionally private.*

Proof: Obvious, since all Bob learns during SETUP and COMMIT is x_1, y_1, and y_0. There is no way to infer x_0 from this information. More precisely, every value in $\mathbf{Z}_p$ is equally likely to be x_0, given what he has seen. If Bob has unlimited powers of computation, it doesn't help him. ∎

Theorem 2 *The proposed commitment scheme is unconditionally binding.*

Proof: After COMMIT, Alice knows a, b, x_0, and y_0, but not Bob's values x_1 and y_1.

Suppose Alice then changes her mind and wishes reveal some value x_0' that is different than x_0. For Bob to accept, she needs to find values x_0', a', and b' such that $y_0 = a'x_0' + b'$ and $y_1 = a'x_1 + b'$. The new line $y = a'x + b'$ must be different than the old line $y = ax+b$, otherwise nothing has changed and she reveals x_0. Either this new line doesn't intersect the old line at all, in which case Bob rejects because (x_1, y_1) should be on the new line, or else the new and old lines intersect at a point (x_2, y_2). Alice only succeeds at cheating if $(x_2, y_2) = (x_1, y_1)$; however, the chance that $(x_1, y_1) = (x_2, y_2)$ is precisely $1/p$, so Alice's chances of cheating are at most $1/p$. ∎

Theorem 3 *Ted never learns the value of x_0.*

Proof: Obvious, since Ted only participates in the SETUP phase. ∎

6 Discussion, Extensions, and Open Problems

6.1 Relation to "Check Vectors"

Our scheme is very close, but not identical, to the use of "check vectors" by Rabin and Ben-Or in their classic paper on multiparty protocols[29]. In their scheme the trusted third party supplies a secret s to Alice and a corresponding check vector to Bob. Later, Alice can forward the secret s to Bob, and Bob can check that Alice has not modified the secret. Our scheme is qualitatively different, because the point here is for Alice to send Bob her own secret, not the trusted third party's secret. Indeed, Alice's secret can be anything, and her secret is not known by the trusted third party at all. But one can also view our scheme as an extension of theirs, since we are effectively using their scheme to reliably transmit (a, b) from Ted to Bob through Alice, but also using (a, b) to allow Alice to commit a new value x_0 to Bob.

6.2 Trusted Initializers

We believe that protocols based on our notion of a "trusted initializer," as formalized here, are worth further exploration. Our notion is somewhat like the notion of a KDC (key distribution center), the notion of having a common random reference string[4], or the notion of the creator of global system parameters, except that our trusted initializer may supply *different, but related* random parameters to each party. It is more like the notion of a "trusted dealer" except that it embodies the restriction that the initialization (deal) should be completed before the other parties have their inputs, and with the restriction that the initializer (dealer) should not participate at all in the subsequent portion of the protocol(s). Where else can trusted initializers be used? Where else have they been used?

6.3 Non-malleability

It is perhaps worth noting that our commitment scheme is "non-malleable" [14]: an adversary intercepting Alice's commitment to Bob can't change it to another commitment to a value x_0' having a known relationship to x_0.

6.4 Zero-knowledge proofs??

One can easily show that any language $L \in NP$ has a perfect zero-knowledge proof in the trusted initializer model; this follows directly from Goldreich et al.[18]. Of course, this is unlikely to be worth bothering about, since trusted initializers were introduced to deal with the case when both Sender and Receiver are both computationally unbounded. For what it is worth, we note that our bit commitments are "chameleon,"—Alice can change her mind if she possesses extra information (what Bob has).

6.5 Multiple "trusted" initializers

If there is no single party that both Alice and Bob trust to serve as a trusted initializer, then they may decide to utilize several trusted initializers, and modify the commitment scheme accordingly. Alice wishes to maintain unconditional privacy of x_0, even if one of the trusted initializers is actually controlled by Bob. Similarly, Bob wishes to maintain unconditional binding, even if one of the trusted initializers is actually controlled by Alice.

For example, one can have unconditional privacy with two trusted initializers, as follows. Alice commits to a random z_1 to Bob using Ted_1 as initializer, and commits $z_2 = z_1 + x_0$ to Bob using Ted_2 as initializer.

Achieving unconditional binding when one of the initializers may be controlled by Alice seems a bit harder, but still doable. They can utilize four initializers as follows. Alice chooses a random value c and defines $z_i = ci + x_0$. She commits to z_i using Ted_i as initializer. After the REVEAL phases are over, Bob can discard any single z_i that Alice has managed to change with the cooperation of her initializer, since that z_i is not on the line formed by the other three. Knowing the correct line allows him to infer x_0.

It is straightforward to generalize these approaches using appropriate secret-sharing schemes, to handle higher thresholds of initializers controlled by Alice or Bob. Suppose that as many as α initializers might be corrupted by Alice, and that as many as β initializers might be corrupted by Bob. Then to preserve unconditional privacy, Alice should define z_i as a random polynomial (with constant term x_0) of degree β in i, instead of the linear polynomial used above. And to prevent Alice from cheating, the total number of initializers should be at least $2\alpha + \beta + 1$. (Analysis details omitted here.)

7 Oblivious Transfer

The notion of oblivious transfer was invented by Rabin [28]; the related notion of a 1-out-of-2 Oblivious Transfer was later devised by Even, Goldreich, and Lempel [15]. There are well-known close connections between commitment schemes and oblivious transfer [22, 27, 10, 12].

We note that a 1-out-of-2 oblivious transfer protocol invented by Bennett et al. [2] for use in a quantum communication model actually works well in our trusted initializer model.

We assume that Alice has two values $m_0, m_1 \in \{0,1\}^k$. The protocol ensures that Bob will obtain m_c, where he gets to choose c ($c = 0$ or 1). But Alice will have no idea as to which message he got, and Bob will have learned nothing about m_{1-c}.

7.1 The BBCS oblivious transfer protocol

The BBCS oblivious transfer protocol is illustrated in Figure 2.
SETUP: Ted privately gives Alice two random k-bit strings r_0, r_1. Ted flips a bit d, and privately gives Bob d and r_d. Ted is now done, and can go home.
REQUEST: Bob determines somehow a bit c; he wants to obtain m_c. He privately sends Alice the bit $e = c \oplus d$.
REPLY: Alice privately sends Bob the values $f_0 = m_0 \oplus r_e$, $f_1 = m_1 \oplus r_{1-e}$. Bob now computes $m_c = f_c \oplus r_d$.

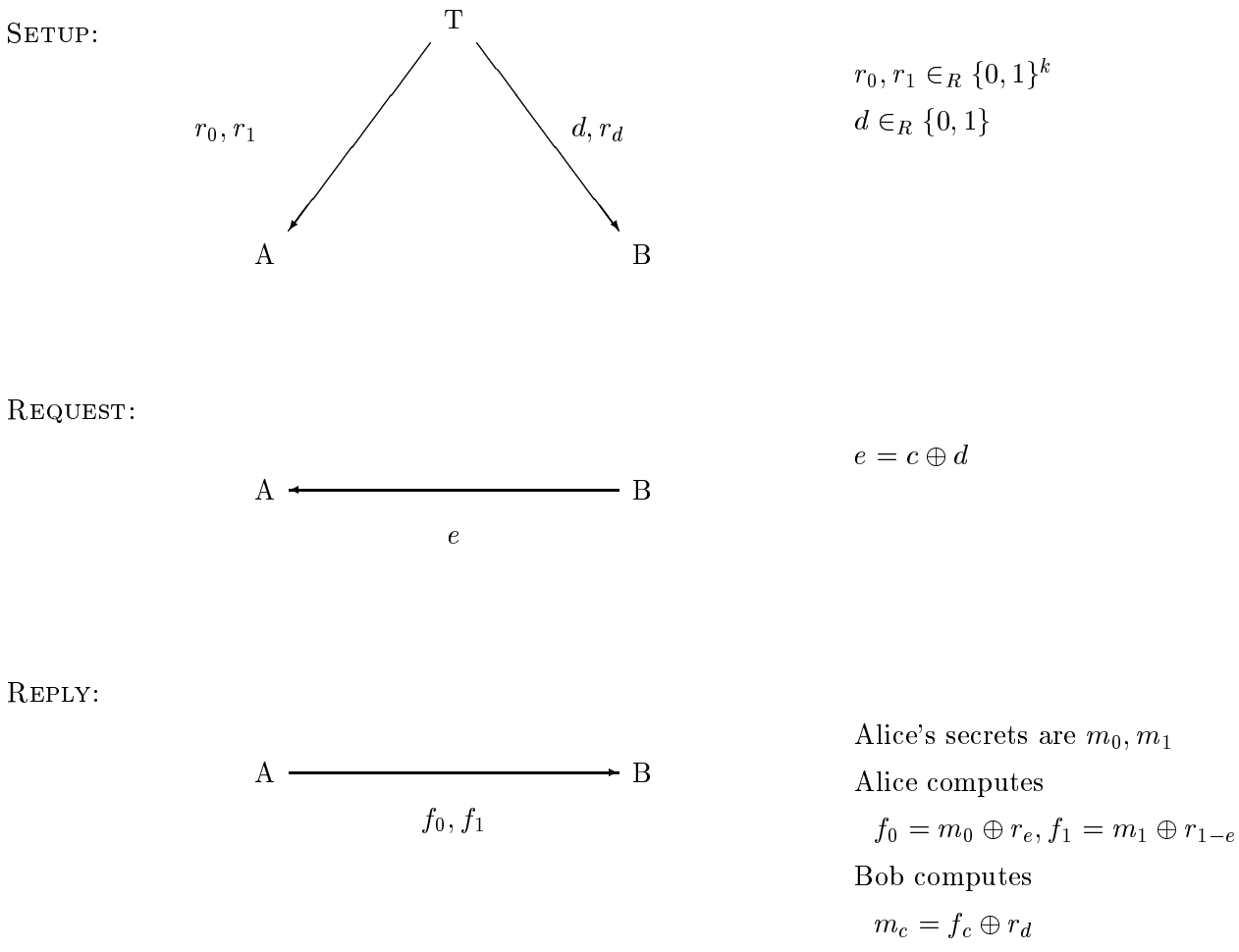


Figure 2: The BBCS oblivious transfer scheme, modified to use a trusted initializer. Alice has two secrets m_0 and m_1 in $\{0,1\}^k$. Bob obtains the value of m_c, where c is his choice. Alice learns nothing about c, and Bob learns nothing about m_{1-c}.

7.2 Analysis and discussion

It is clear that Alice has no information about c, and that Bob has no information about m_{1-c}. The scheme clearly generalizes to 1-out-of-n in an easy manner, using n random strings $r_1, r_2, \ldots, r_n$.

Acknowledgments

I would like to thank Victor Boyko,Shafi Goldwasser, Stas Jarecki, Silvio Micali, and Zulfikar Ramzan for helpful discussions.

References

[1] Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. Multi-prover interactive proofs: How to remove intractability assumptions. In *Proc. 20th Annual ACM Symposium on Theory of Computing*, pages 113–132. ACM, 1988.

[2] Charles H. Bennett, Gilles Brassard, Claude Crèpeau, and Marie-Hélène Skubiszewska. Practical quantum oblivious transfer. In J. Feigeanbaum, editor, *Proc. CRYPTO 91*, pages 351–366. Springer, 1992. Lecture Notes in Computer Science No. 576.

[3] M. Blum. Coin flipping by telephone. In *Proc. IEEE Spring COMPCOM*, pages 133–137. IEEE, 1982.

[4] M. Blum, P. Feldman, and S. Micali. Non-interactive zero-knowledge and its applications. In *Proc. 20th Annual ACM Symposium on Theory of Computing*, pages 103–112. ACM, 1988.

[5] G. Brassard, D. Chaum, and C. Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

[6] G. Brassard, C. Crèpeau, R. Jozsa, and D. Langlois. A quantum bit commitment scheme provably unbreakable by both parties. In *Proceedings of the 34th Annual IEEE Symposium on the Foundations of Computer Science*, pages 362–371, Nov 1993.

[7] G. Brassard and M. Yung. One-way group actions. In A.J. Menezes and S. A. Vanstone, editors, *Proc. CRYPTO 90*, pages 94–107. Springer-Verlag, 1991. Lecture Notes in Computer Science No. 537.

[8] Gilles Brassard, Claude Crépeau, and Moti Yung. Constant-round perfect zero-knowledge computationally convincing protocols. *Theoretical Computer Science*, 84(1):23–52, 22 July 1991.

[9] C. Crèpeau and J. Kilian. Achieving oblivious transfer using weakened security assumptions. In *Proc. 29th FOCS Conference*, pages 42–52. IEEE, 1988.

[10] Claude Crépeau. Efficient cryptographic protocols based on noisy channels. In Walter Fumy, editor, *Proc. EUROCRYPT 97*, pages 306–317, Paris, 1997. Springer. Lecture Notes in Computer Science No. 1233.

[11] Ivan Damgård. On the existence of bit commitment schemes and zero-knowledge proofs. In G. Brassard, editor, *Proc. CRYPTO 89*, pages 17–27. Springer-Verlag, 1990. Lecture Notes in Computer Science No. 435.

[12] Ivan Damgård, Joe Kilian, and Louis Salvail. On the (im)possibility of basing oblivious transfer and bit commitment on weakened security assumptions. In Jacques Stern, editor, *Proc. CRYPTO 99*, pages 56–73. Springer, 1999. Lecture Notes in Computer Science No. 1592.

[13] Ivan Damgård, Torben P. Pedersen, and Birgit Pfitzmann. On the existence of statistically hiding bit commitment schemes and fail-stop signatures. *Journal of Cryptology*, 10(3):163–194, 1997.

[14] D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. In *Proc. STOC '91*, pages 542–552. ACM, 1991.

[15] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. In R. L. Rivest, A. Sherman, and D. Chaum, editors, *Proc. CRYPTO 82*, pages 205–210, New York, 1983. Plenum Press.

[16] A. Fiat and A. Shamir. How to prove yourself: practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *Proc. CRYPTO 86*, pages 186–194. Springer, 1987. Lecture Notes in Computer Science No. 263.

[17] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proc. 19th ACM Symposium on Theory of Computing*, pages 218–229. ACM, 1987.

[18] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(1):691–729, 1991.

[19] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Computing*, 17(2):281–308, April 1988.

[20] Shai Halevi. Efficient commitment schemes with bounded sender and unbounded receiver. In Don Coppersmith, editor, *Proc. CRYPTO 95*, pages 84–96. Springer, 1995. Lecture Notes in Computer Science No. 963.

[21] Shai Halevi and Silvio Micali. Practical and provably-secure commitment schemes from collision-free hashing. In Neal Koblitz, editor, *Proc. CRYPTO 96*, pages 201–215. Springer, 1996. Lecture Notes in Computer Science No. 1109.

[22] Joe Kilian. Founding cryptography on oblivious transfer. In *Proc. 20th Annual ACM Conference on Theory of Computing*, pages 20–31. ACM, 1988.

[23] H.-K. Lo and H. F. Chau. Is quantum bit commitment really possible?, Mar 1996. Preprint archive `http://xxx.lanl.gov/ps/quant-ph/9603004`.

[24] D. Mayers. Unconditionally secure quantum bit commitment is impossible. *Physical Review Letters*, 78:3414–3417, 1997.

[25] Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 2(2):151–158, 1991.

[26] Kazuo Ohta, Tatsuaki Okamoto, and Atsushi Fujioka. Secure bit commitment function against divertibility. In R.A. Rueppel, editor, *Advances in Cryptology — Eurocrypt '92*, pages 324–340, Berlin, 1992. Springer-Verlag.

[27] Rafail Ostrovsky, Ramarathnam Venkatesan, and Moti Yung. Secure commitment against a powerful adversary. In A Finkel and M. Jantzen, editors, *Proceedings of STACS 92*, pages 439–448. Springer-Verlag, 1992. Lecture Notes in Computer Science No. 577.

[28] M. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory, 1981.

[29] Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proc. 21st Annual ACM Symposium on Theory of Computing*, pages 73–85, May 1989.

[30] Louis Salvail. Quantum bit commitment from a physical assumption. In Hugo Krawczyk, editor, *Proc. CRYPTO 98*, pages 338–353. Springer-Verlag, 1998. Lecture Notes in Computer Science No. 1462.