Fischer's Cryptographic Protocols

Rebecca N. Wright
Computer Science Department
Stevens Institute of Technology
Hoboken, NJ 07030
rwright@cs.stevens-tech.edu

ABSTRACT

This note is prepared for Michael Fischer's 60th birthday celebration at PODC 2003. In it, I briefly describe some of Michael Fischer's work on distributed cryptographic protocols.

1. INTRODUCTION

Although Michael Fischer is best known for his seminal work in distributed consensus and his early work in complexity theory, he has also made a number of contributions at the intersection of cryptography and distributed computing—cryptographic protocols.

In this note, I overview Fischer's work on three types of cryptographic protocols: oblivious transfer, electronic voting, and key exchange. This is a brief and informal overview; the interested reader is invited to consult the full papers and the references therein for more information.

2. OBLIVIOUS TRANSFER

Oblivious transfer, introduced by Rabin [8], is a mechanism by which Alice, holding a message, can interact with Bob in such a way that with probability 1/2, Bob receives Alice's message, and with probability 1/2, Bob learns nothing at all about the message. Alice should have no information about which of these two events occurred. Furthermore, even a cheating Alice should not be able to prevent Bob from learning the message with probability at least 1/2 without Bob detecting her cheating, and if Bob follows his protocol, a cheating Alice should not learn anything about Bob's outcome. Similarly, a cheating Bob should have probability no more than 1/2 of learning anything about Alice's message if Alice is following her protocol. (For a thorough description and more formal treatment, see [8, 4].)

Although it is hard to imagine a scenario in which Alice and Bob's actual goal would be to execute an oblivious transfer, it has turned out to be extremely useful as a primitive in numerous secure distributed protocols.

Appeared in *Proceedings of 22nd Annual ACM Symposium on Principles of Distributed Computing (PODC)*, 2003.

In his seminal work [8], Rabin demonstrated an oblivious transfer protocol based on the difficulty of factoring. However, Rabin's protocol has not been proven correct, in the sense that even if the assumption that it is difficult to factor is true, it could turn out that Bob could learn more information about the message than he is supposed to. Fischer, Micali, and Rackoff [4] showed how to modify Rabin's protocol so that it can be proven correct based only on the assumption that factoring is hard.

RABIN'S PROTOCOL [8]. In this description, we assume that Alice and Bob both know a number m, the product of two large primes, and the message that Alice has and wishes to obliviously transfer to Bob is the factorization p, q of m.

1. Bob chooses a random x between 1 and m that is relatively prime to m, computes $y = x^2$, and sends y to Alice.

2. Alice, knowing the factorization of m, computes z, a random square root mod m of y, and sends z to Bob. (If no square root exists, Alice halts immediately.)

3. Bob checks that $z^2 = y \bmod m$. If not, Bob states that Alice is cheating. Otherwise, z is a square root of y, one of $\{x, -x, w, -w\}$ for some $w \neq x$. If $z = x$ or $z = -x$, then Bob has not learned anything, as he already knew x. If $z = w$ or $z = -w$, then Bob can compute $p = \gcd(m, x - z)$ and $q = m/p$, and output p, q as the factorization of m.

Clearly, if Alice and Bob both follow the protocol, Bob learns the factorization of m with probability exactly 1/2 and Alice does not know whether this occurred or not. It is also straightforward to see that a cheating Alice cannot reduce Bob's probability of learning p and q, since she does not know which square root x Bob used to compute y. It is also straightforward that a cheating Bob cannot gain an advantage by sending a y that is not a square mod m or by choosing an x in some clever way instead of randomly. However, the possibility remains open that Bob might be able to send a square y for which he does not know any square roots of y, and thereby gain an advantage. That is, although no such algorithms are known to exist, it could be possible that there are both a polynomial-time algorithm that, given m, produces a square y with high probability, and a polynomial-time algorithm that, given m, y, and z such that $z^2 = y \bmod m$, factors m with high probability.

In the absence of showing that such algorithms do not exist, the fix for Rabin's protocol is to add a step before

Alice provides a square root to Bob in which Bob proves to Alice, using a "cut-and-choose" zero-knowledge proof, that he knows a square root of $y \bmod m$, without revealing any information to Alice about which square root he knows.

Fischer, Micali, and Rackoff's subprotocol [4]. Assume that Bob knows x, a square root of $y \bmod m$. The even number n is a security parameter, in that choosing n larger reduces Bob's chance of successfully cheating.

1a. Bob chooses n random numbers $r_1, \ldots, r_n$, each between 1 and m and relatively prime to m. For each i, Bob computes $y_i = yr_i^2$ and sends it to Alice.

1b. Alice chooses a random subset T consisting of half of the y_i's sent by Bob, and sends it back to Bob.

1c. Bob checks that what Alice sent is properly constructed (i.e. that it is a subset consisting of half of the values he sent her). If not, he outputs "cheating" and halts. Otherwise, for each y_i in T, he sends r_i to Alice, and for each y_i not in T, he sends $z_i = xr_i$.

1d. Alice checks the values sent by Bob to see that for each y_i in T, $y_i = yr_i^2$ and for each y_i not in T, $y_i = z_i^2$. If the values pass the check, Alice is convinced and continues to step 2 of the Rabin protocol. Otherwise, she outputs "cheating" and halts.

Intuitively, Alice is correct to be convinced if the values sent by Bob pass the check, because in this case it is extremely likely that there is at least one i for which he knows both r_i and z_i such that $y_i = yr_i^2$ and $y_i = z_i^2$, and so he knows the square root $z_i/r_i \bmod m$ of y. If there is not such an i, there is at most one choice of T for which the check would be passed, so Bob is extremely likely to be caught cheating.

With the addition of the subprotocol, the revised Rabin protocol can now be proven secure assuming it is computationally infeasible to factor the product of two large primes. That is, if Bob could cheat Alice with non-negligible probability, he could also factor m with non-negligible probability without any interaction with Alice at all.

3. ELECTRONIC VOTING

Electronic voting has been a very hot area in recent years. Along with early work of Chaum [1], some of the earliest work on secure electronic voting was due to Cohen (now Benaloh) and Fischer [2]. Benaloh and Fischer introduced the notion of an electronic voting scheme in which, using cryptographic techniques, voters and outside observers could verify that the tally of the election was computed correctly without needing to or being able to learn the actual votes themselves. In their work, the election authority *can* learn how the individual voters voted. However, their ideas were some of the initial seeds that have been incorporated into many current voting schemes that use additional techniques to provide complete privacy of the voters' votes, preventing the election authority or authorities from learning voters' individual votes.

More specifically, Benaloh and Fischer present an electronic voting scheme in which, under assumptions about the difficulty of certain number-theoretic problems, the following properties hold with high probability:

- each voter can check whether the outcome of the election was computed properly (even if the election authority is cheating),
- the privacy of each voter's vote is protected against any coalition of dishonest voters (beyond what the coalition can infer from the tally and their own votes), and
- no coalition of dishonest voters can prevent the successful completion of the election.

Benaloh and Fischer election protocol [2]. The basic paradigm of their protocol for a yes-no election is a four phase process:

1. The election authority selects and reveals a set of parameter values that will be used in the election, and interactively proves that the parameters were chosen correctly according to certain specifications.
2. Each voter selects and reveals an unmarked ballot consisting of one encrypted "yes" vote and one encrypted "no" vote in random order, and an interactive proof that the ballot is of this form.
3. Each voter then selects one encrypted "yes" or "no" vote from her ballot, according to her desired vote.
4. The election authority computes and announces the tally, (but not the actual votes), and engages in an interactive proof that the tally was computed correctly.

The specifics of phases 2 and 4 use similar interactive proof techniques as the oblivious transfer in Section 2; many other details must also be addressed. The reader is referred to their paper [2].

4. KEY EXCHANGE

Motivated by the game of bridge, Fischer, Paterson, and Rackoff [3] consider the following problem: Alice and Bob have been dealt hands from a shuffled deck of cards; any remaining cards go to an eavesdropper, Eve. Can Alice and Bob communicate in Eve's presence to determine a bit b that Alice and Bob know but Eve does not? They show that the answer is yes in some cases, such as when Alice and Bob jointly hold at least two more cards than Eve, and no in other cases, such as when Alice, Bob, and Eve each hold only a single card.

The problem and results were extended by Fischer and Wright [5, 6, 7] (in work that eventually led to my Ph.D. under Fischer's supervision) to n-bit key exchange in a multiparty setting in which a *team* of players $P_1, \ldots, P_k$ holds hands of $h_1, \ldots, h_k$ cards, respectively, and Eve holds any remaining e cards.

A *key set* is a set $\{x, y\}$ of two cards such that P_i holds one and P_j holds one, but Eve does not know which party holds which card. Once P_i and P_j have determined a key set they hold, they can use it to transmit a secret bit b from P_i to P_j. Namely, P_i says (randomizing the order of the two parts of her statement: "if I hold x, then the secret bit is b; if I hold y, then the secret bit is $1 - b$."

The simplest protocol is a one-bit secret key exchange protocol called the "key set protocol." The idea is for pairs of players to find key sets they hold until all the players are connected by a spanning tree of key sets. Once this happens, a designated player can choose a random bit b as the secret key and it can then be communicated secretly to all players over the key set links as described above. We say a player P_i is *feasible* if $h_i > 1$ or if $h_i = 1$ and $e = 0$.

FISCHER AND WRIGHT KEY SET PROTOCOL. [5]

1. Let P be the feasible player holding the fewest cards. If no player is feasible, then P is the lowest-numbered player holding a non-empty hand, if there is one.

2. P chooses a random card x contained in her hand and a random card y not in her hand. She proposes $K = \{x, y\}$ as a key set by asking, "Does any team player hold a card in K?"

3. If another team player Q holds y, she knows that K is a key set, so she accepts K by announcing that she holds a card in K. The cards x and y are discarded without revealing which player holds which card (except to be used later for the secret transmission of the eventual key bit). Whichever player of P and Q holds fewer cards exposes the remaining cards in her hand, which are discarded, and drops out of the protocol. The remaining team players go back to step 1 with the resulting new deal.

4. If none of the team players holds y, then K is rejected. In this case, x and y are discarded, and the players go back to step 1.

Execution continues in this manner until either there are not enough cards left to complete steps 1 and 2 (in which case the protocol fails), or until only one team player is left. In the latter case, all team players are now connected by key sets, and can exchange a secret bit as previously described. An inductive argument shows this protocol works if $\max h_i + \min h_i \geq k + e$.

5. CONCLUSIONS

Michael Fischer has had an impact on theoretical computer science that is both broad and deep. As my thesis advisor, he has also had a significant impact on my life. I thank him for both.

6. REFERENCES

[1] D. Chaum. "Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms," *Communications of the ACM*, Vol. 24, No. 2, 1981, pp. 84–88.

[2] J. Cohen (Benaloh) and M. J. Fischer, "A Robust and Verifiable Cryptographically Secure Election Scheme", *Proceedings of 26th Symposium on Foundations of Computer Science (FOCS)*, 1985, pp. 372–382.

[3] M. J. Fischer, M. S. Paterson, and C. Rackoff, "Secret Bit Transmission Using a Random Deal of Cards," *Distributed Computing and Cryptography*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Volume 2, American Mathematical Society, (1991), pp. 173–181.

[4] M. J. Fischer, S. Micali, and C. Rackoff, A Secure Protocol for the Oblivious Transfer, *J. Cryptology*, Vol. 9, No. 3, Summer 1996, pp. 191–195. (Originally presented at Eurocrypt '84, but not published in proceedings.)

[5] M. J. Fischer and R. N. Wright, "Multiparty Secret Key Exchange Using a Random Deal of Cards," *Advances in Cryptology – Proceedings of Crypto '91*, Springer-Verlag, LNCS 576, 1992, pp. 141–155.

[6] M. J. Fischer and R. N. Wright, "An Efficient Protocol for Unconditionally Secure Secret Key Exchange," *Proceedings of the 4th Annual Symposium on Discrete Algorithms (SODA)*, 1993, pp. 475–483.

[7] M. J. Fischer and R. N. Wright, "Bounds on Secret Key Exchange Using a Random Deal of Cards," *J. Cryptology*, Vol. 9, No. 2, Spring 1996, pp. 71–99.

[8] M. O. Rabin, "How to Exchange Secrets by Oblivious Transfer," Technical Report TR-81, Aiken Computation Lab, Harvard University (1981).