Multiparty Secret Key Exchange Using a Random Deal of Cards[1]

(Extended Abstract)

Michael J. Fischer
Computer Science Department
Yale University
New Haven, CT 06520–2158
fischer-michael@cs.yale.edu

Rebecca N. Wright
Computer Science Department
Yale University
New Haven, CT 06520–2158
wright-rebecca@cs.yale.edu

Abstract

We consider the problem of *multiparty secret key exchange*. A "team" of players P_1 through P_k wishes to determine an n-bit secret key in the presence of a computationally unlimited eavesdropper, Eve. The team players are dealt hands of cards of prespecified sizes from a deck of d distinct cards; any remaining cards are dealt to Eve. We explore how the team can use the information contained in their hands of cards to determine an n-bit key that is secret from Eve, that is, an n bit string which each team player knows exactly but for which Eve's probability of guessing the key correctly is $1/2^n$ both before and after she hears the communication between the team players. We describe randomized protocols for secret key exchange that work for certain classes of deals, and we present some conditions on the deal for such a protocol to exist.

1 Introduction

An important problem of cryptography is the problem of *multiparty secret key exchange*. This can be viewed as a multiparty protocol between a group of players. At some point, a subset of $k \geq 2$ players P_1 through P_k form a *team*. The rest of the players are considered eavesdroppers. The team players carry out randomized algorithms. Each player's random choices are private to that player. All communication is by public broadcast and is overheard by the eavesdroppers. The following scenario demonstrates a situation in which the need for secret key exchange might arise.

A certain government agency handles security of information on a "community of interest" basis. For each project within the agency, a group of people are chosen to work on the project. We call this group a team. Teams form and dissolve as various projects are started and completed. All communication regarding the project is intended to be shared with those on the team, and to be kept secret from those outside the team. However, the security of the various communication channels—the telephone, interoffice mail, electronic mail, and face-to-face communication—is not guaranteed. Hence, each team that forms would like to exchange a secret key, which it can then use as a part of

[1]This research was supported in part by National Science Foundation grant IRI–9015570.

Appears in *Proceedings of Crypto '91*, Springer-Verlag, LNCS 576, (1992)

some cryptographic protocol to securely send all further communication regarding the project. Another place where this problem may arise is in a distributed system, for example a computer network linking a corporation's headquarters and branch offices.

Formally, the team wishes to determine a random n-bit sequence S satisfying agreement, secrecy, and uniformity. *Agreement* is met if each team player knows S. *Secrecy* is met if the eavesdroppers' probability of guessing S correctly is the same before and after hearing the communication between the team players. *Uniformity* requires that S has equal probability of being any one of the 2^n possible n-bit sequences. Such a secret key is said to be *shared* by the team. Each team player has an output tape that is physically protected from the other players. An *n-bit secret key exchange protocol* is one in which each team player outputs the same n-bit sequence satisfying the secrecy and uniformity conditions. The output can then be used for a variety of cryptographic purposes, for example, as the key in private key cryptosystems (cf. [DH]).

We allow the eavesdroppers to be computationally unlimited, so standard cryptographic techniques based on computational difficulty cannot be used. In fact, a secret key exchange protocol is not possible without any further assumptions, for an eavesdropper can simulate any team player under all possible random choices and thereby learn S. Hence, we give the players secret initial information in the form of correlated random variables. While the value of each player's random variable is unknown to the other players, the distribution from which the random variables are chosen is publicly known. For any team that forms, the remaining players are assumed to collaborate against the team, possibly communicating among themselves via private channels. Thus we treat them as a single eavesdropper, Eve, who possesses the initial information of all of the non-team players. Note that because initial information is given to all players before the team forms, it is not possible to deny Eve all initial information. We would like to distribute the initial information in such a way that any team that forms can obtain a secret key.

Our framework is very general and admits the trivial solution in which each player is given *a priori* a secret key for each team to which the player might eventually belong. Any team that forms can use the corresponding preassigned secret key, but since there is an exponential number of possible teams, the amount of initial information is quite high. Also, the structure of the initial random information is rather complicated.

We desire instead correlated random variables that have a simple structure and a small amount of initial information. A familiar example of such correlated random variables is provided by ordinary card games in which players are dealt hands from a randomly shuffled deck of cards. By looking at her own cards, a player gains some information about the other players' hands. Namely, she learns a set of cards that appear in no other player's hand. Peter Winkler developed bidding conventions for the game of bridge whereby one player could send her partner secret information about her hand that was totally unrelated to the actual bid and completely undecipherable to the opponents, even though the protocol was known to them [Fl, Wi81a, Wi81b, Wi83]. Fischer, Paterson and Rackoff [FPR] carried this idea further, using deals of cards for secret bit transmission between two players. We consider secret key exchange protocols based on such card games in the remainder of this paper.

The problem of secret key exchange has been considered by others in the context of public key cryptography (cf. [DH, Me]). Impagliazzo and Rudich provide evidence that most of the standard techniques in cryptography cannot be used to construct a secret key exchange protocol from a one-way permutation [IR]. Our results are quite different in character from these, for we place no computational limitations on our participants. Thus, one-way permutations do not exist in our model, and one must rely on other assumptions, such as the existence of prior secret initial information as in this paper, in order to make the problem solvable. Furthermore, techniques such as those used by Maurer [Ma] will not work here since we require that the key obtained is *completely* secret from Eve and is known *exactly* to all the team players, as prescribed by the secrecy and agreement conditions.

In the remainder of the paper, we consider the situation in which a team has just formed, and investigate whether secret key exchange is possible. We use the following terminology. A *deck* D is a finite set, whose elements we call *cards*; a *hand* is subset of D. Let d be the size of the deck. The cards in the deck are known to all the players, as is the size of each player's hand, but the cards in each player's hand are private to that player. In an $(h_1, h_2, \ldots, h_k; e)$-*deal*, each team player P_i is given a hand H_i such that $H_i \subseteq D$ and $|H_i| = h_i$. Eve is dealt a hand E such that $E \subseteq D$ and $e = |E| = d - \sum_{i=1}^{k} h_i$. The deal $\delta = (H_1, H_2, \ldots, H_k; E)$ is *legal* if $H_1, H_2, \ldots, H_k, E$ partition D. We call the description of the sizes of the hands, $\xi = (h_1, h_2, \ldots, h_k; e)$, the *signature*[2] of the deal, and call a deal having signature ξ a ξ-*deal*. If all k team players have the same hand size h in a signature, we write $(h^k; e)$.

An n-bit secret key exchange protocol that always succeeds in obtaining an n-bit secret key for all legal ξ-deals is said to *work for* ξ. We also say such a protocol *performs* *n-bit secret key exchange for* ξ.

In Section 2, we describe a simple 1-bit secret key exchange protocol that works for all deals in which the team players' hands are sufficiently large relative to the size of

[2]This term is borrowed from algebra, and is not intended to have any connection to digital signatures.

the team and the size of Eve's hand. In Section 3, we present a protocol that improves on the first protocol in two ways. First, it establishes an n-bit secret key for arbitrary n. Second, it requires only that each team player hold an arbitrarily small fraction of the cards (assuming that the deck is sufficiently large). In Section 4, we present some necessary conditions on the deal for a secret key exchange protocol to exist. In Section 5, we show that the protocol presented in Section 2 is optimal for a natural class of related protocols.

2 A One-Bit Secret Key Exchange Protocol

We first consider a simple 1-bit secret key exchange protocol. We use the notion of a *key set* defined in [FPR]. A key set K consists of two cards, one held by a team player P, the other held by a *different* team player Q. A key set $K = \{x, y\}$ is *opaque* if, given the information available to Eve, it is equally likely that P holds x and Q holds y or that P holds y and Q holds x.

Once P and Q determine an opaque key set K that they hold, they can use it to obtain a bit r that is secret to Eve. Namely, they agree that $r = 0$ if P holds x and $r = 1$ if P holds y, or vice versa. Thus K acts as a 1-*bit secret channel*; that is, it allows P and Q to communicate a single bit secretly.

The structure of our protocol is as follows. We think of the team players as nodes of a graph. We connect two team players by an edge if the team players have a 1-bit secret channel between them. The goal of the protocol is to connect the team players. We obtain 1-bit secret channels by finding opaque key sets between pairs of team players until the team is connected. Then a designated player, say P_1, chooses a bit s randomly. Using flooding on the 1-bit secret channels, s is propagated to all the team players. Clearly s satisfies agreement and uniformity. Secrecy is satisfied because each 1-bit channel preserves secrecy. Hence, s is a 1-bit secret key.

We define the notion of a feasible player. Let each team player P_i hold h_i cards and let Eve hold e cards. Then P_i is *feasible* if $h_i > 1$, or if $h_i = 1$, $e = 0$, and $h_j > 1$ for all $j \neq i$. In the protocol that follows, we say a card x is *discarded* from the deck if all team players agree to play as if x is no longer part of the deck. Similarly, we say a team player P *drops out* of the protocol if the team players agree to play as if P were no longer part of the team. The protocol follows.

1. Let P be the feasible player holding the smallest hand. (Ties are broken in favor of

the lower-numbered player.) If no player is feasible, then P is the lowest-numbered player holding a non-empty hand, if any.

2. P chooses a random card x contained in her hand and a random card y not in her hand and proposes $K = \{x, y\}$ as a key set by asking, "Does any team player hold a card in K?"[3]

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. 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 "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.

The execution of the protocol continues in this manner until either there are not enough cards left to complete steps 1 and 2, or until only one team player is left. In the first case, the protocol fails. In the second case, all the team players are connected by opaque key sets. To see this, note that every key set $K = \{x, y\}$ accepted in step 3 is opaque because it is equally likely to be proposed by P in the symmetric deal where everything is the same except that P holds y and Q holds x. Hence the team can obtain a 1-bit secret key by flooding as previously described. We call this protocol the SFP key set protocol (for smallest feasible player). An inductive argument shows the following.

Theorem 2.1 *Let $\xi = (h_1, ..., h_k; e)$. Let $h_i \geq 1$ for $1 \leq i \leq k$, and $\max h_i + \min h_i \geq k + e$. Then the SFP key set protocol performs 1-bit secret key exchange for ξ.*

In Section 5 we consider protocols with different rules for choosing P in step 1. We show there that the SFP key set protocol is optimal among all such key set protocols.

3 An n-Bit Secret Key Exchange Protocol

The SFP key set protocol has two limitations: it requires that the team hold more than half the cards in the deck, and it only provides a 1-bit secret key. Moreover, it is not obvious how to modify the protocol to overcome these limitations. For example, the

[3] In an abstract setting, $\{x, y\}$ is clearly the same as $\{y, x\}$. In an actual implementation, care must be taken that the communication of $\{x, y\}$ does not reveal which card came from P's hand.

protocol cannot be repeated to obtain additional key bits since players drop out and expose their remaining cards during execution.

The first limitation is overcome in [FPR] for a team of two players. A 1-bit secret key exchange protocol is presented there that works when each team player holds any fixed fraction of the cards and the deck is sufficiently large. An analysis of that protocol establishes the following:

Theorem 3.1 (Fischer, Paterson, Rackoff) *There is a 1-bit secret key exchange protocol $\mathcal{P}$ such that for all $0 < \beta \leq 1/2$ and $d \geq \left(\frac{2}{\beta^2}\right) 2^{1/\beta}$, $\mathcal{P}$ works for $(\lfloor \beta d \rfloor, \lfloor \beta d \rfloor; d - 2\lfloor \beta d \rfloor)$.*

We show how to use such a protocol to perform n-bit secret key exchange for teams of size k and sufficiently large decks. Our construction is a general reduction of the n-bit, k-player problem for signature $\xi^* = (h^k; d - kh)$ to the 1-bit, 2-player problem for signature $\xi = (\lfloor h/2n \rfloor, \lfloor h/2n \rfloor; d - 2\lfloor h/2n \rfloor)$. Thus, given a protocol $\mathcal{P}$ that performs 1-bit secret key exchange for ξ, we construct a new protocol $\mathcal{P}^*$ that performs n-bit secret key exchange for ξ^*.

Lemma 3.1 *Let $n \geq 1$, $k \geq 2$ and $d \geq kh$. Let $\mathcal{P}$ be a 1-bit secret key exchange protocol that works for*

$$\xi = \left(\left\lfloor \frac{h}{2n} \right\rfloor, \left\lfloor \frac{h}{2n} \right\rfloor; d - 2\left\lfloor \frac{h}{2n} \right\rfloor \right).$$

Then there is a protocol $\mathcal{P}^*$ that performs n-bit secret key exchange for $\xi^* = (h^k; d - kh)$.

Proof: Suppose n, k, d, h, $\mathcal{P}$, and ξ satisfy the conditions of the lemma, and let $\xi^* = (h^k; d - kh)$. We construct an n-bit secret key exchange protocol $\mathcal{P}^*$ that works for ξ^*.

Assume the players are linearly ordered, say, by their indices. Two team players are said to be *neighbors* if they are adjacent in the ordering. P_1 is the leader and randomly chooses an n-bit string S to be the secret key. Each pair of neighbors P_i and P_{i+1} uses $\mathcal{P}$ in sequence n times to establish an n-bit secret key B_i that they share, as described in detail below. When P_i learns S from P_{i-1}, she sends $E_i = S \oplus B_i$ to P_{i+1} publicly. P_{i+1} recovers S by computing $E_i \oplus B_i$.

We now describe in detail how the one-time pads are established. Given a team player P_i, we say P_{i+1} is the *right neighbor* of P_i and P_{i-1} is the *left neighbor* of P_i. Each player P_i divides her hand into $2n$ parts, H_i^1 through H_i^{2n}, of size $\lfloor h/2n \rfloor$ and a (possibly empty) part containing her remaining cards. P_i uses parts H_i^1 through H_i^n to establish B_i with her right neighbor, and she uses parts H_i^{n+1} through H_i^{2n} to establish B_{i-1} with her left neighbor.

The j^{th} bit of the one-time pad B_i is gotten as follows. P_i plays the role of player 1 in $\mathcal{P}$, pretending that the only cards she holds are those in H_i^j. P_{i+1} plays the role of player 2 in $\mathcal{P}$, pretending that the only cards she holds are those in H_{i+1}^{n+j}. The other team players do not participate. We call the cards in $H_i^j \cup H_{i+1}^{n+j}$ the *current cards*. Both players pretend that Eve holds all but the current cards. Thus P_i and P_{i+1} execute $\mathcal{P}$ as if the deal were a ξ-deal. Since $\mathcal{P}$ is assumed to work for ξ, P_i and P_{i+1} obtain a shared secret bit, which they use for the j^{th} bit of B_i.

Note that whenever a card x not in the current cards is referenced, all players behave as if Eve holds x. If Eve does not hold x, she learns that x does not lie in the current cards, but she learns nothing further about the location of x. Thus this process can be repeated, using each part of each team player's hand exactly once, to get all the one-time pads. ∎

We now apply Lemma 3.1 to families of 1-bit protocols.

Theorem 3.2 *Let $n \geq 1$, $k \geq 2$, and let f be a function on the reals. Suppose for every $0 < \beta \leq 1/4$ and every $d \geq f(\beta)$ that there is a 1-bit secret key exchange protocol $\mathcal{P}$ that works for $(\lfloor \beta d \rfloor, \lfloor \beta d \rfloor; d - 2\lfloor \beta d \rfloor)$. Let $0 < \alpha \leq 1/k$, and let $d \geq f(\alpha/2n)$. Let $\mathcal{P}^*$ be the protocol constructed as in the proof of Lemma 3.1. Then $\mathcal{P}^*$ performs n-bit secret key exchange for $(\lfloor \alpha d \rfloor^k; d - k\lfloor \alpha d \rfloor)$.*

Proof: Assume the hypotheses of the protocol, and assume we are given a deal of signature $\xi = (\lfloor \alpha d \rfloor^k; d - k\lfloor \alpha d \rfloor)$. Let $h = \lfloor \alpha d \rfloor$ and let $\beta = \alpha/2n$. Since $\alpha \leq 1/k$, it follows that $d \geq k\lfloor \alpha d \rfloor = kh$ and $\beta \leq 1/4$. Also, since n is an integer, $\lfloor \beta d \rfloor = \lfloor \alpha d/2n \rfloor = \lfloor \lfloor \alpha d \rfloor /2n \rfloor = \lfloor h/2n \rfloor$. Hence, $\mathcal{P}$ satisfies the conditions for Lemma 3.1. It follows from Lemma 3.1 that $\mathcal{P}^*$ performs n-bit secret key exchange for $(h^k; d - kh) = \xi$ as desired.
∎

The following corollary to Theorem 3.2 is immediate using Theorem 3.1, taking $f(\beta) = \left(\frac{2}{\beta^2}\right) 2^{1/\beta}$.

Corollary 3.1 *Let $0 < \alpha \leq 1/k$. Suppose $d \geq 8\left(\frac{n}{\alpha}\right)^2 2^{2n/\alpha}$. Then $\mathcal{P}^*$ performs n-bit secret key exchange for $(\lfloor \alpha d \rfloor^k; d - k\lfloor \alpha d \rfloor)$.*

Unfortunately, the required deck size here grows exponentially in n/α. Richard

Beigel [Be] has suggested an improved 1-bit two-player protocol in which the deck size appears to grow only polynomially in $1/\alpha$. Using such a protocol, our construction yields an n-bit team protocol for which the deck grows only polynomially in n/α.

4 Lower Bound Results

In order to discuss lower bounds, we first define our model more precisely. We look at a synchronous distributed model of computation in which there is a *team* of k players P_1 through P_k and a passive eavesdropper, Eve. Let $\mathcal{P}$ be an n-bit secret key exchange protocol for P_1 through P_k. In each round of $\mathcal{P}$, each of the team players simultaneously broadcasts a message to all of the other players. All messages are overheard by Eve. Let Z be the set of possible messages, and let $z_i \in Z$ be the message that each P_i sends in the round. The k-tuple $(z_1, z_2, \ldots, z_k) \in Z^k$ is called a *statement* of $\mathcal{P}$. A sequence of statements is called a *conversation* of $\mathcal{P}$, denoted by $\tau_{\mathcal{P}}$. We assume each protocol $\mathcal{P}$ always terminates after some fixed number $t_{\mathcal{P}}$ of rounds. A conversation $\tau_{\mathcal{P}}$ is *complete* if $|\tau_{\mathcal{P}}| = t_{\mathcal{P}}$. As it will be clear from context which protocol is being discussed, we will omit the protocol subscripts.

The protocol run by each player P_i is a randomized algorithm that determines the message for P_i to send at each round based on her hand and the conversation so far. Specifically, let $\mathcal{H}_i$ be the set of possible hands for P_i. Let $H_i \in \mathcal{H}_i$, and let σ be a conversation. A protocol for P_i is a pair $(\mu_i, \mathcal{O}_i)$. If σ is not complete, $\mu_i(H_i, \sigma)$ is a random variable over the message space Z, where $\Pr[\mu_i(H_i, \sigma) = z]$ is the probability that P_i sends message z at round $r+1$ given that P_i holds hand H_i and the conversation through round r is σ. If σ is complete, $\mathcal{O}_i(H_i, \tau) \in \{0,1\}^n$ specifies P_i's output value.

A *joint protocol* for players P_1 through P_k consists of a set of protocols $(\mu_i, \mathcal{O}_i)$, where each $(\mu_i, \mathcal{O}_i)$ is a protocol for P_i. All the protocols $(\mu_i, \mathcal{O}_i)$ are known to each team player, as well as to Eve. Thus an n-bit secret key exchange protocol that works for ξ is a joint protocol $\{(\mu_1, \mathcal{O}_1), \ldots, (\mu_k, \mathcal{O}_k)\}$ for the team players such that for all possible runs on each legal ξ-deal, if every team player P_i plays according to $(\mu_i, \mathcal{O}_i)$, the team succeeds in obtaining an n-bit secret key. It is a straightforward exercise to modify the protocols we describe in English in this paper to fit this model.

We generalize a theorem of [FPR] to show that secret key exchange is not possible if the deal does not provide sufficient shared information. Throughout the remainder of this section, we fix a deck D and a signature $\xi = (h_1, h_2, \ldots, h_k; e)$ such that $\sum_{i=1}^{k} h_i + e = |D|$.

Recall that a ξ-deal of a deck D is a collection of $k+1$ hands $(H_1, \ldots, H_k; E)$ such that $|H_i| = h_i$ for $i \in \{1, \ldots, k\}$ and $|E| = e$, and recall that a deal is legal if the hands partition D. We sometimes use the term "general deal" to refer to a deal that is not necessarily legal. Let Δ' be the set of all (general) ξ-deals of D, and let Δ be the set of legal ξ-deals of D. Note that $\Delta \subseteq \Delta'$ and that a general deal δ is legal if and only if the hands in δ are pairwise disjoint.

A random legal deal is a uniformly distributed random variable over Δ. A random general deal is a uniformly distributed random variable over Δ'. Note that in both a random legal deal and in a random general deal, each hand H_i is uniformly distributed over $\mathcal{H}_i$. The difference is that in a random general deal, the hands $H_1, \ldots, H_k$ are independent random variables, whereas in a random legal deal, they are correlated. Hence, only in a random legal deal does player P_i get any information about the cards in other player's hands.

Let $\overline{\gamma}$ be the probability that a random general deal is also a legal deal. Intuitively, the smaller $\overline{\gamma}$ is, the more shared information the deal contains. The following theorem provides an upper bound on $\overline{\gamma}$ in order for n-bit secret exchange to be possible.

Theorem 4.1 *Let ξ and $\overline{\gamma}$ be as defined above, and let $n \geq 1$. If $\overline{\gamma} > 1/2^{k-1}$, then no protocol performs n-bit secret key exchange for ξ.*

Proof (sketch): Assume to the contrary that some n-bit secret key exchange protocol works for ξ when $\overline{\gamma} > 1/2^{k-1}$. We may assume without loss of generality that $n = 1$. Using a somewhat involved probabilistic argument, we show that $|\Delta|/|\Delta'| \leq 1/2^{k-1}$, i.e., at most $1/2^{k-1}$ of all deals are legal. Since all deals are equally likely, it follows that $\overline{\gamma} \leq 1/2^{k-1}$, a contradiction. We conclude that no protocol performs 1-bit secret key exchange for ξ. ∎

The full proof is rather long and is omitted. (It may be found in [FW].) We remark that the theorem holds even for protocols in which Eve is not allowed to look at her hand. Thus, our theorem applies to a larger class of protocols than necessary. We do not know how to use Eve's ability to see her cards to improve this result.

Corollary 4.1 *Let $n \geq 1$ and $2 \leq k \leq 8$. Then no protocol performs n-bit secret key exchange for $(1^k; 1)$.*

Proof: In these cases, $\overline{\gamma} = (k+1)!/(k+1)^k > 1/2^{k-1}$. ∎

For $k > 8, \overline{\gamma} = (k+1)!/(k+1)^k < 1/2^{k-1}$, so nothing can be concluded.

Theorem 4.1 says nothing about the $(1^k;0)$ case. However, it is possible to show the following.

Theorem 4.2 *Let $n \geq 1$. Then no protocol performs n-bit secret key exchange for $(1,1,1;0)$.*

Proof (sketch): It is sufficient to show no protocol performs 1-bit secret key exchange for $(1,1,1;0)$. To prove this, we look at properties of the possible conversations of a 1-bit secret key exchange protocol on $(1,1,1;0)$-deals. Let τ be a complete conversation. We say that τ is *realizable* if there is some $\delta \in \Delta$ such that τ is a possible conversation of the protocol when the deal is δ, and in this case we say δ is *consistent* with τ. An output $v \in \{0,1\}$ is *possible* given τ if there is some $\delta = (H_1, H_2, H_3) \in \Delta$ consistent with τ such that $v = \mathcal{O}_i(H_i, \tau)$ for each i.

Suppose $\mathcal{P}$ performs 1-bit secret key exchange for $(1,1,1;0)$. We construct a tree of conversations as follows. The nodes of the tree are conversations, and the edges out of a node are labeled by possible next statements. Thus the interior nodes are partial conversations; leaf nodes are complete conversations. A conversation τ *passes through* a node σ if τ extends σ. It can be shown that exactly two deals are consistent with each realizable conversation, and that both of the deals consistent with a realizable conversation have the same parity[4]. We say that the parity of a realizable conversation τ is the parity of the two deals consistent with τ. We say a node is *single valued* if all conversations passing through it have the same parity. It is *multivalued* otherwise. We are now ready to derive a contradiction.

By the correctness of $\mathcal{P}$, all $(1,1,1;0)$-deals must be possible initially. Thus the root of the tree is multivalued. Because only one conversation passes through any leaf node, all leaves are single valued. Hence there must be a multivalued node σ having only single valued children. Thus there exist complete conversations τ_0 and τ_1 passing through σ such that τ_0 has parity 0 and τ_1 has parity 1. It is then possible to construct an "interpolated" conversation passing through σ that gives rise to a multivalued child, a contradiction. ∎

This proof is highly dependent on specific properties of the set of possible $(1,1,1;0)$-deals, and does not generalize easily to larger teams. However, using an extension to the graph theoretical framework developed by Beaver, Haber and Winkler [BHW] to represent shared knowledge between two players, it is possible to show the following general result (cf. [FWW]).

[4] The parity of a $(1,1,1;0)$-deal is the parity of the permutation describing it.

Theorem 4.3 *Let $n \geq 1$, $k \geq 2$, and $e \geq 0$. Then no protocol performs n-bit secret key exchange for $(1^k; e)$ unless $n = 1, k = 2$, and $e = 0$.*

5 Key Set Protocols Revisited

Even for the simple case of $n = 1$, there is a large gap between signatures for which we have a secret key exchange protocol and signatures for which we have shown that no protocol exists. For example, $(2, 2, 2; 2)$ falls into this gap.

One approach to closing the gap is to modify the SFP key set protocol presented in Section 2. In step 1 of this protocol, a team player P, *the proposer*, is chosen. By considering different rules for choosing the proposer, we get a class of protocols. We call such a rule a *proposing rule*. We require a proposing rule to be a deterministic function of the current signature. We call the protocol that results from proposing rule $\mathcal{R}$ the *$\mathcal{R}$ key set protocol*. We call the class of all such protocols the class of key set protocols. By this definition, the SFP key set protocol results from the *smallest feasible player* proposing rule (SFP): If any team player is feasible, the feasible player with the smallest hand is chosen. (Ties are broken in favor of the lower-numbered player.) If no team player is feasible, the lowest-numbered team player holding a non-empty hand is chosen, if any.

Theorem 2.1 holds for any $\mathcal{R}$ key set protocol where $\mathcal{R}$ always chooses a feasible player if some team player is feasible. The converse, however, does not in general hold. For example, the signature $\xi = (3, 3, 2, 1; 1)$ does not satisfy the conditions of the theorem, but the SFP key set protocol works for ξ. We have been unable to find an exact characterization of the signatures for which the SFP key set protocol works. Nevertheless, it is possible to show that the SFP key set protocol is optimal for the class of key set protocols. By this we mean that for a signature ξ, if the $\mathcal{R}$ key set protocol works for ξ for some $\mathcal{R}$, then the SFP key set protocol also works for ξ. To prove this we look at a simple combinatorial stick game between a team and an adversary. The stick game abstracts the important aspects of the key set protocol.

The stick game is a game between a team and an adversary. There are k team piles, P_1 through P_k, and a pile E. Pile P_i contains h_i sticks, and pile E contains e sticks. The team always moves first. On the team's turn, the team designates a team pile P_i containing at least one stick. On the adversary's turn, the adversary either removes one stick from P_i and one from E (allowed only when $e > 0$), or chooses another team pile P_j such that $h_j > 0$, removes the smaller of P_i and P_j entirely, and removes one stick from the larger pile. Note that removing a pile is not the same as removing all the sticks in the pile. Play ends when there are one or zero team piles, in which case the team wins, or when there is no move available (either to the team or to the adversary), in which case the team loses. A configuration of the stick game can be described by the tuple $(h_1, \ldots, h_k; e; I)$, where I specifies whether it is the team's turn (T) or the adversary's turn (A). We call the stick game starting from configuration C the C *stick game*.

A *strategy* for the team, or team strategy, is a function that, given a configuration of the stick game where it is the team's turn specifies the next team move. Similarly, an adversary strategy is a function that specifies the next adversary move. We say a configuration C is *winning* if there is some team strategy $\mathcal{S}$ such that if the team plays the C stick game by strategy $\mathcal{S}$, then the team wins regardless of the moves chosen by the adversary. We say $\mathcal{S}$ is a *successful team strategy for* C. We call $\mathcal{S}$ an *optimal team strategy* if it is a successful team strategy for every winning configuration C. We similarly define *optimal adversary strategy*.

The stick game is a finite game, since every adversary turn decreases the total number of sticks by at least two. Furthermore, it is a game of complete information, since the team and the adversary take turns and all information about the state is known to both the team and the adversary. Hence game theory tells us that every configuration is either winning or losing, and an optimal team strategy $\mathcal{S}$ and an optimal adversary strategy $\mathcal{A}$ both exist [BCG].

We define a feasible pile in a stick game configuration exactly as we defined a feasible player in a signature, and we similarly define the SFP strategy for the team in the stick game. It is easy to see that a configuration in the stick game is winning for a given team strategy if and only if the key set protocol works for the corresponding signature when the team plays according to the corresponding proposing rule. Hence to show the optimality of the corresponding SFP key set protocol we need only show the optimality of the SFP stick game strategy.

We show this by a series of arguments known as strategy stealing arguments. We define $\text{size}((h_1, \ldots, h_k; I)) = k + e$. The strategy stealing arguments are by induction on $\text{size}(C)$. We construct configurations $C_1, \ldots, C_i$ and $C'_1, \ldots, C'_j$ as shown in Figure 1.

$$\begin{array}{ccc}
C_0 \text{ winning} & & C'_0 \text{ winning} \\
\Downarrow & & \Uparrow \\
\vdots & & \vdots \\
\Downarrow & & \Uparrow \\
C_i \text{ winning} & \Longrightarrow & C'_j \text{ winning}
\end{array}$$

Figure 1: The strategy stealing argument.

The configurations $C_1, \ldots$ are constructed by playing the C_0 stick game. We assume the team never makes a move that would take a winning configuration to a losing one, and we specify the adversary moves. Since an adversary move cannot take a winning configuration to a losing one, it follows that if C_0 is winning, then every C_ℓ is winning. Similarly, the configurations $C'_1, \ldots$ are constructed by playing the C'_0 stick game. We assume the adversary never makes a move on a losing configuration that results in a winning configuration, and we specify the team moves. It follows that if C'_0 is losing, then every C'_ℓ is losing, or conversely, if any C'_ℓ is winning, then C'_0 is winning. The construction terminates when we obtain C_i and C'_j for which we can show that if C_i is winning then C'_j is also winning.

A case by case analysis of possible adversary responses to each SFP team move enables us to prove the following. (A full proof appears in [FW]).

Theorem 5.1 *The SFP strategy is an optimal team strategy for the stick game, and hence the SFP key set protocol is optimal for the class of key set protocols.*

Theorem 5.1 indicates that changing the proposing rule is not a sufficient modification to the key set protocol to close the gap described at the beginning of the section. However, there are other possible modifications to the key set protocol to consider. For example, one might allow the players to communicate in order to choose the proposer. This also does not close the gap, for we can show that the SFP key set protocol is optimal for the larger class of protocols this gives rise to. However, the optimality may fail if the proposed key set is allowed to be chosen non-randomly.

In the key set protocols described here, every time a key set is found, one of the team players discards all the cards in her hand and drops out of the protocol, except to wait to hear the secret bit. We do this in order to avoid getting more than one key set between any two players. It would be possible to consider key set protocols in which a team player only drops out when a team player in the same connected component of the key set graph is chosen to propose a key set. We suspect that this does not give the team additional power, and conjecture that Theorem 5.1 holds for this larger class of protocols.

Another possible modification to the key set protocol is to allow team players to discard only the key set cards and risk getting multiple key sets between two team players. It is an open question whether multiple key sets can be used (for example to "send" some of the cards in a player's hand to another player) to achieve 1-bit secret key exchange where no key set protocol of the class described in this paper succeeds.

6 Concluding Remarks

We have shown here some conditions on the signature of the deal that allow secret key exchange and some conditions under which secret key exchange is not possible. However, there is a large gap. There are many signatures for which we can neither give a secret key exchange protocol nor demonstrate the nonexistence of such a protocol.

As a future direction for this work, we intend to look at the concept of shared secret information between a team. We would like to develop a theory of shared secret information which can be applied to arbitrary correlated random variables. Specifically, can we quantify how many bits of shared secret information a deal contains for the team? How can we use this information to develop better protocols and tighter lower bounds on the signatures for which secret key exchange is possible? More generally, what other mechanisms besides deals from a common deck of cards give correlated random variables that can be used for secret key exchange?

Deals of cards have a small amount of initial information. However, deals of cards appear somewhat inefficient for secret key exchange, in that the number of secret bits the team can obtain is small in comparison to the number of cards they are dealt. Michael Rabin [Ra] suggests a protocol that uses private correlated random variables to solve another classical security problem, authentication. His method requires random variables that appear to contain more initial information than a deal of cards, but also appear to contain more shared secret information. We would like to use the theory of shared secret information suggested above to quantify the ratio of initial information to shared secret information, and to investigate upper and lower bounds on this ratio for secret key exchange protocols.

7 Acknowledgements

We thank Michael Merritt for his contribution to the proof of Lemma 3.1. We thank Peter Winkler for many helpful comments. We thank Nick Reingold for countless discussions, and for suggesting a simpler proof of a key lemma used in the full proof of Theorem 4.1.

References

[BHW] D. Beaver, S. Haber, and P. Winkler. On the Isolation of a Common Secret, preprint, Bellcore, 1991.

[Be] R. Beigel. 1991. (Private communication.)

[BCG] E. R. Berlekamp, J. H. Conway, and R. K. Guy. *Winning Ways*, Volume I, Academic Press Inc., London, 1982.

[DH] W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Trans. Inform. Theory, IT–22, Vol. 6,* Nov. 1976, pp. 644–654.

[FPR] M. J. Fischer, M. S. Paterson, and C. Rackoff. Secret Bit Transmission Using a Random Deal of Cards, *Distributed Computing and Cryptography*, American Mathematical Society, 1991, pp. 173–181.

[FWW] M. J. Fischer, P. Winkler, and R. N. Wright. June 1990. (Private communication.)

[FW] M. J. Fischer and R. N. Wright. Multiparty Secret Key Exchange Using a Random Deal of Cards, *Technical Report YALEU/DCS/TR–855,* Yale University, June 1991.

[Fl] J. Flint. Cheating by Degrees, *The Times Saturday Review*, May 9, 1981.

[IR] R. Impagliazzo and S. Rudich. Limits on the Provable Consequences of One-way Permutations, *Proc. 21st ACM Symposium on Theory of Computing*, May 1989, pp. 44–61.

[Ma] U. M. Maurer. Perfect Cryptographic Security from Partially Independent Channels, *Proc. 23rd ACM Symposium on Theory of Computing*, May 1991, pp. 561–571.

[Me] R. C. Merkle. Secure Communication over Insecure Channels, *Comm. ACM 21,* 4, April 1978, pp. 294–299.

[Ra] M. Rabin. Cryptography Without Secrets. Presented at *DIMACS 1990 Workshop on Cryptography*, Princeton, NJ. October 1–4, 1990.

[Wi81a] P. Winkler. Cryptologic Techniques in Bidding and Defense, Parts I, II, III, and IV, *Bridge Magazine*, April—July, 1981.

[Wi81b] P. Winkler. My Night at the Cryppie Club, *Bridge Magazine*, August 1981, pp. 60–63.

[Wi83] P. Winkler. The Advent of Cryptology in the Game of Bridge, *Cryptologia*, Vol. 7, No. 4, October 1983, pp. 327–332.