The Zero-Knowledge Match Maker $\ddot\smile$

Dedicated to Arjeh Cohen on the occasion of his retirement

Tom Verhoeff, Dept. of Math & CS, Eindhoven University of Technology

18 June 2014

Abstract

Berry Schoenmakers recently showed me some of the fun things he does in his crypto courses. One of these things is his implementation of the protocol to do a secure computation of the product of two secret bits. In this article, I present my variant of Berry's implementation.

1 Bit Products and Match Making

More popularly, computing the bit product is known as *match making*, where two parties privately indicate whether they like the other party (the two secret bits), and then the protocol determines whether or not the liking is mutual. Under this protocol, only the outcome is revealed, and no other information except what is derivable from the outcome. In particular, when the liking is not mutual, the outcome does not reveal how it was brought about, that is, who did not like the other.

2 Bert den Boer's Five-Card Trick

The protocol invented by Bert den Boer of CWI [1] is also known as the *Five-Card Trick.* Alice and Bob each get two cards of the same type, say an Ace of Hearts (♡) and an Ace of Clubs (♣). The protocol proceeds as follows.

1. Alice and Bob each make their choice in private, and encode it with the cards:
   - Alice indicates that she likes Bob by stacking the ♡ face down *on top of* the face-down ♣.
   - Bob does the same thing, except that he puts ♡ *below* ♣ to indicate he likes Alice.
2. The cards are now put together:
   - First, Alice' card pair (still face down),
   - then another (neutral) face-down ♡ is put on top of that, and
   - finally Bob's card pair is put on top of that (still face down).
3. A secret random cyclic permutation (also known as a *cut*) of the closed five-card deck is applied. That is, no one knows which cut was applied. For instance, Alice and Bob each apply a random cut to the deck.

4. The card deck is inspected. There are only two possible card sequences modulo a cyclic permutation (also see Figure 1):

    - Either three cyclicly adjacent ♡ and two cyclicly adjacent ♣ appear,
    - or two cyclicly adjacent ♡ and two isolated ♣, sandwiching the third ♡.

    The first order corresponds to a match. Note that these two patterns are invariant under reversal of the deck.

Alice	**Bob**	**Deck**
Yes	Yes	♣ ♡ ♡ ♡ ♣
Yes	No	♣ ♡ ♡ ♣ ♡
No	Yes	♡ ♣ ♡ ♡ ♣
No	No	♡ ♣ ♡ ♣ ♡

Figure 1: All possible deck orders under den Boer's protocol

3 Berry's Implementation

Berry lets Alice and Bob place their cards on a turntable (see Figure 2). The random cyclic permutation (cut) is obtained by spinning the turntable, like a wheel of fortune.

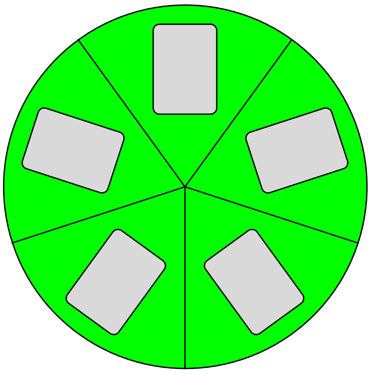

Figure 2: Berry's turntable with five face-down cards

Berry's implementation still has some practical shortcomings.

1. For one thing, you need three decks of playing cards, to obtain five cards suitable for the protocol.

2. All cards need to be placed identically on the turntable, because otherwise you might reconstruct the permutation.

3. The cards somehow need to be secured to the turntable to provide a centripetal force, for otherwise they would fly off during the spinning.

4 My Variant

I set out to design something to overcome the shortcomings of Berry's implementation. My first thought was to address shortcoming 1, by creating some custom cards and (2D) print them on stiff paper. At the same time, I could choose other designs for the card faces, to make the rules for Alice and Bob more intelligible (see Figure 3). To indicate a like, Alice places her cards to construct an arrow pointing toward Bob. Likewise, Bob makes an arrow pointing toward Alice for a like.

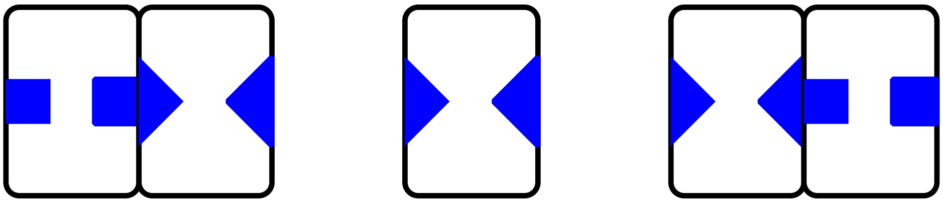

Figure 3: First design for custom cards, based on arrows

To address the other two shortcomings (of freedom in card placement and of secure attachment to the turntable), I decided to create cards in the shape of circular sectors (see Figure 4 and 7). That way, they could snugly fit into compartments of the turntable, leaving no freedom of placement. Figure 5 shows the match and no-match patterns.

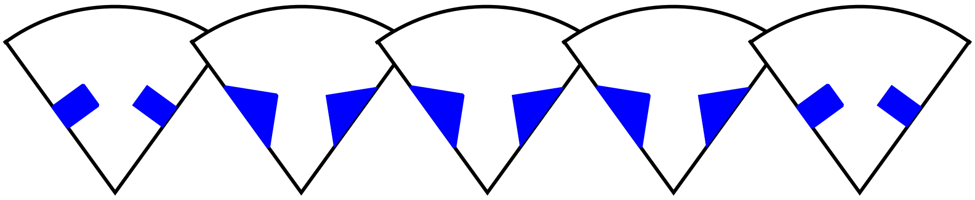

Figure 4: Second design for custom cards, based on arrows and circular sectors

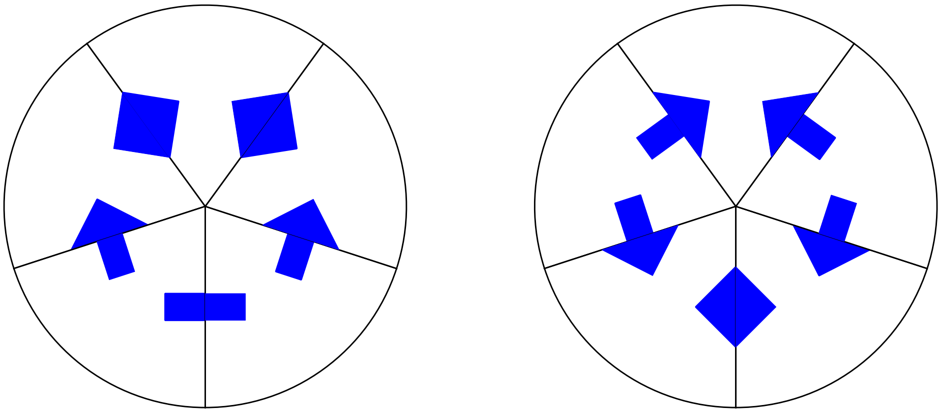

Figure 5: Patterns for a match (left) and no match (right) with arrow-sector cards

After taking a second look at the match pattern in Figure 5, I could not resist seeing a smiley, but with a wry look on its face. Therefore, I adapted the design somewhat, sacrificing the clarity of the arrows for a better smiley (see Figures 6).

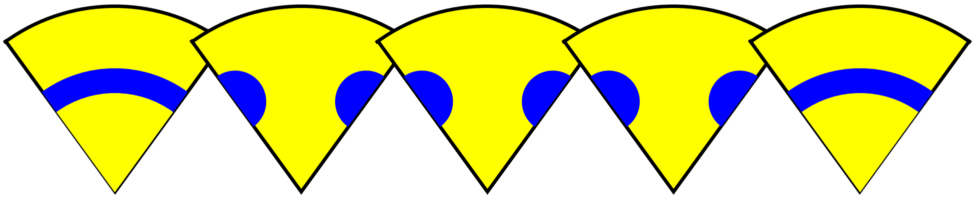

Figure 6: Third design for custom cards, based on circular sectors of a smiley

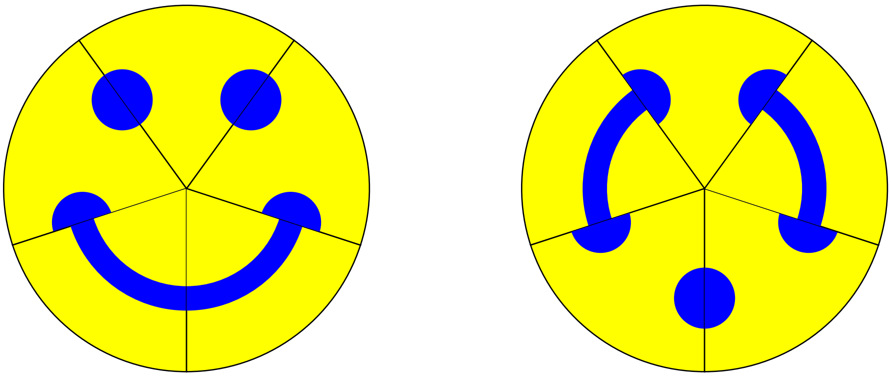

Figure 7: Patterns for a match (left) and no match (right) with smiley-sector cards

5 Conclusion

An interactive demonstration is available in [2]. The next step is to turn these designs into 3D-printable objects, that can be snapped together to form a spinning top, which can then be spun on the table for randomization.

With the help of some serendipity, I came up with an elegant design in the form of smiley sectors to overcome the shortcomings of Berry's implementation of den Boer's five-card trick.

I would like to thank Berry Schoenmakers for introducing me to his implementation of the five-card trick, and for the ensuing discussions. As future work, we will be looking for ways of incorporating this variant of the protocol into some kind of social board game.

References

[1] B. den Boer. "More Efficient Match-Making and Satisfiability: The Five Card Trick", *Advances in Cryptology – EUROCRYPT '89*, Lecture Notes in Computer Science, Volume 434, 1990, pp.208–217.

[2] T. Verhoeff. "Zero-Knowledge Match Maker", June 2014. `www.win.tue.nl/~wstomv/misc/ZeroKnowledgeMatchMaker.html`. Submitted to *Wolfram Demonstrations Project*.