

# Efficient Correlated Action Selection

**Mikhail Atallah, Marina Blanton, Keith Frikken, and Jiangtao Li**

Department of Computer Science  
Purdue University

Financial Cryptography and Data Security  
(FC'06)

February – March 2006

# Introduction

- We consider a game-theoretic problem of *two player strategic games*.
- In such games, each user has a set of possible moves, and both players execute their moves simultaneously.
- There is a *payoff* function which is computed on the two moves.
- It is assumed that both players are *selfish and rational*, i.e., want to maximize their expected payoff.
- A *strategy* for a player is a (possibly randomized) method for choosing a move.

## Introduction (cont.)

- It has been shown in the game theory literature that higher payoffs can be achieved if the players coordinate their actions.
  - such strategies are called correlated.
- To implement this, a trusted third party mediator performs action selection for the participants and privately tells each player what its designated move is.
- The players are incentivized to follow the recommendation.
- The moves can be chosen according to a probability distribution.

## An Example of Correlated Strategy

- Consider two competing stores selling secondhand furniture from failed dot-coms.
- Each week each of the stores has to decide whether to run a sale or not.
- Each of them must choose in advance.
- Possible outcomes:
  - both decide to keep regular prices (acceptable)
  - one runs a sale (acceptable)
  - both run a sale (unacceptable)

## An Example of Correlated Strategy (cont.)

- The payoffs and probabilities can look like:

	No sale	Sale
No sale	9, 9	5, 12
Sale	12, 5	0, 0

	No sale	Sale
No sale	5/11	3/11
Sale	3/11	0

- The problem: potentially beneficial collaborations do not take place because of the fear that the players' private information might be misused.
- This is where cryptographic techniques come handy.

## Problem Description

- Consider a two-party game, where two entities want to coordinate their respective actions.
- The joint strategy is described by a list of  $m$  pairs.
- Each pair has a certain probability of being chosen.
- A pair of actions is chosen randomly according to this probability distribution.
- Each player learns its respective move and nothing else.

## Background

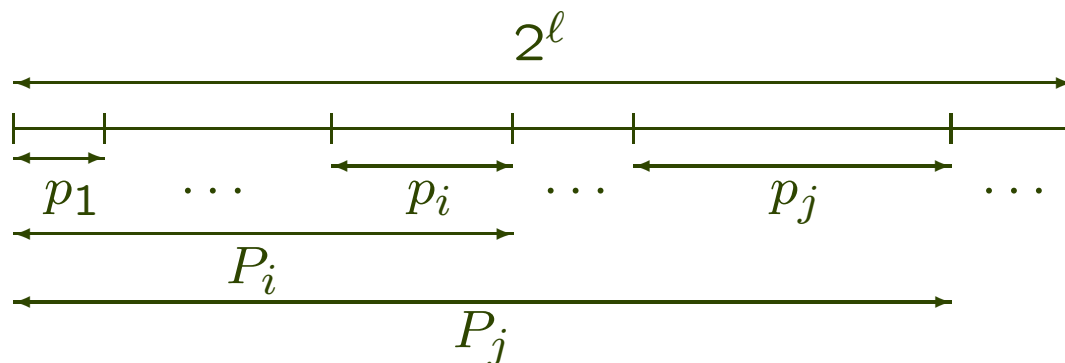
- Dodis, Halevi, and Rabin (CRYPTO'00) eliminated the need for a third-party mediator.
  - their solution is efficient, but assumes a uniform distribution.
  - it becomes inefficient when the probabilities vary.
- Teague (FC'04) subsequently extended this work to non-uniform distributions.
  - her solution performs better when the probabilities significantly vary.
  - but it is still worst-case exponential in the representation of the joint strategy.
- Our approach is more efficient than these and circuit simulation approaches.

## Notation

- The  $m$  action pairs are denoted as  $\{(a_i, b_i)\}_{i=1}^m$ .
- Each pair can be chosen with probability  $q_i$ , with the sum of all of them being 1.
- We convert each  $q_i$  into its integer representation  $p_i$  of  $\ell$  bits.
- Without loss of generality, let  $\sum_{i=1}^m p_i = 2^\ell$  (or else pad the list with a dummy pair).
- Now we can refer to the problem description as  $m$  tuples  $(a_i, b_i, p_i)$ .

## High Level Description of the Solution

- Let's call the first player Alice and the second player Bob.
- Alice and Bob jointly compute  $P_i = \sum_{j=1}^i p_j$  for  $1 \leq i \leq m$ .



- They also generate a random number  $r \in [0, 2^\ell - 1]$ .
- Note that the probability that  $r \in [P_{i-1}, P_i)$  is  $p_i/2^\ell$ .
- All that Alice and Bob need to do is to find the index  $i$  such that  $r < P_i$  and  $r \geq P_{i-1}$  and obtain  $a_i$  and  $b_i$ , respectively.

## Semi-Honest Protocol at High Level

- We use a semantically secure homomorphic encryption scheme (Paillier).
- One player (Alice) generates a key pair  $(pk, sk)$ , the second player (Bob) has access only to the public key.
- An interesting building block is a binary search protocol.
  - it searches on an array of additively split data items.
  - the outcome of the search (i.e., the index) becomes known to both players.
  - this doesn't compromise the security, but allows for a more efficient solution.

## Semi-Honest Protocol (cont.)

- The protocol steps:
  - Each player in turn blinds and permutes encrypted tuples  $\{(\text{Enc}_{pk}(a_i), \text{Enc}_{pk}(b_i), \text{Enc}_{pk}(p_i))\}_{i=1}^m$ .
  - They compute the encryptions  $\text{Enc}_{pk}(P_i)$  using the permuted values.
  - They jointly generate  $r \xleftarrow{R} \{0, 1\}^\ell$ .
  - They additively split (in modular arithmetic) the  $P_i$ 's and run a binary search protocol to determine index  $j$  such that  $P_{j-1} \leq r < P_j$ .
  - Alice recovers  $a_j$ , and Bob recovers  $b_j$ .
- The protocol's complexity is  $O(m + \ell \log m)$ .

## Handling Dishonest Behavior

- It would be inefficient to make the preceding solution secure against malicious behavior.
  - the nature of the steps involved would require very expensive zero-knowledge proofs.
- Instead, we give a new protocol based on the same general idea.
- Tools used:
  - threshold (2,2) homomorphic ElGamal encryption.
  - two-party computation based on the conditional gate (Schoenmakers and Tuyls, ASIACRYPT'04).
- The overall protocol has complexity  $O(ml)$ .

## Handling Dishonest Behavior (cont.)

- Additional sub-protocols are:
  - Addition of bitwise-encrypted values
    - uses conditional gates.
    - computes exclusive OR and majority functions.
  - Constant round comparison protocol
    - utilized conditional gates.
  - Binary search protocol
    - the main idea is the same as in the semi-honest setting.
    - uses the above comparison protocol as a subroutine.

## Comparison with Prior Work

- Comparison of worst case performance (computation and communication):

	Teague	SFE	Our Protocols
semi-honest	$O(\max\{m, 2^\ell\})$	$O(m\ell)$	$O(m + \ell \log m)$
malicious	$O(\sigma \cdot \max\{m, 2^\ell\})$	$O(m\ell)$	$O(m\ell)$

- $m$  is the number of action pairs.
- $\ell$  is the number of bits representing the probabilities.
- $\sigma$  is a security parameter for the cut-and-choose technique (must be linear in the payoffs to prevent cheating).

## Conclusions

- We gave a secure protocol for correlated action selection which is more efficient than previous results and has important applications in game theory.
- Our protocol in the malicious setting is linear in the input size, while the protocol in the semi-honest setting is sub-linear.
- It is an interesting research problem to narrow the gap in the complexities between these two models.